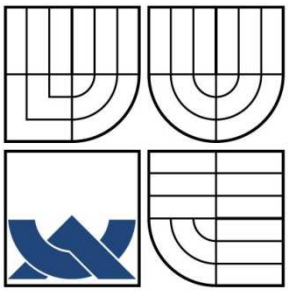
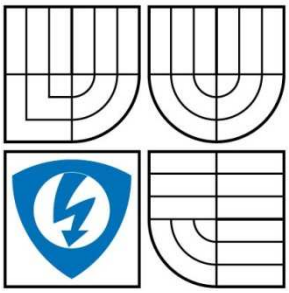


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

NEBEZPEČÍ INTERNETOVÉ KOMUNIKACE

INTERNET COMMUNICATION RISKS

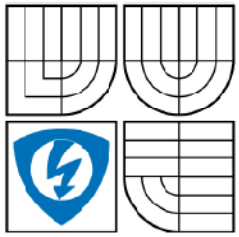
BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE — RADEK ŠTĚRBA
AUTHOR

VEDOUCÍ PRÁCE — ING. MICHAL POLÍVKA
SUPERVISOR

BRNO 2009



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student: Radek Štěrba ***ID:*** 72859
Ročník: 3 ***Akademický rok:*** 2008/2009

NÁZEV TÉMATU:

Nebezpečí internetové komunikace

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s protokolem TCP/IP v souvislosti s protokoly pro rychlou výměnu zpráv (tzv. instant messaging - IM) a hlasovou komunikací po IP síti (VoIP). Prostudujte problematiku IM protokolů, technologii adresářových služeb (např. LDAP) a problematiku softwarových (VoIP) ústředen. Vybudujte jednotnou databázi uživatelů, sdílenou mezi systémem IM a hlasovanými účastníky (VoIP) komunikace. Na serverovou část řešení navažte odpovídající část klientskou (IP telefony, IM klienty,...) tak, aby umožňovala přímo využívat vybudované databáze. Navržený jednotný komunikační systém by měl celkově zefektivnit komunikaci uvnitř organizace.

DOPORUČENÁ LITERATURA:

[1] DOSTÁLEK, Libor, KABELOVÁ, Alena. Velký průvodce protokoly TCP/IP a systémem DNS. 3. vyd. Praha: Computer Press, 2002. 542 s. ISBN 80-7226-675-6.
[2] SCAMBRAY, Joel, MCCLURE, Stuart, KURTZ, George. Hacking bez tajemství. Praha: Computer Press, 2001. 592 s. ISBN 80-7226-549-0.
[3] CHALUPA, Radek. 1001 tipů a triků pro Visual C++. Praha: Computer Press, 2003. 436 s. ISBN 80-7226-842-2.
[4] VIRIUS, Miroslav. Jazyky C a C++. Praha: [s.n.], 2005. 520 s. ISBN 80-247-1494-9.

Termín zadání: 9.2.2009 ***Termín odevzdání:*** 2.6.2009

Vedoucí práce: Ing. Michal Polívka

prof. Ing. Kamil Vrba, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

Anotace

Tato práce se zabývá problematikou komunikačních protokolů a serverů. Jsou zde popsány protokoly TCP/IP. Dále jsou popsány čtyři nejpoužívanější IM protokoly (popis přihlašování, komunikace a zabezpečení). Jsou zde popsány protokoly pro hlasovou komunikaci a problematika softwarových ústředen. Dále jsou zde nastíněny databázové systémy hlavně LDAP.

Součástí této práce je také praktická činnost – zprovoznění serveru pro hlasovou komunikaci a posílání zpráv a databáze LDAP. Je popsán postup pro zprovoznění jednotlivých serverů, navázání příslušných klientů na servery a v neposlední řadě komunikace s LDAP.

Klíčová slova

TCP/IP, Instant Messaging, Jabber, VoIP, SIP, LDAP, server, klient, odposlech, ústředna.

Abstrakt

This work deals with problems of communications protocols and servers. Primarily protocols TCP/IP are described. Next four the most used IM protocols (account of log in, communications and security) are described. Protocols for voice communications and problems of software phone systems are described here. Also database systems especially LDAP are mentioned here.

A practical working is a part of this work too – a putting server for voice communications and sending messages and database LDAP into service. The process for putting the servers into service, connecting competent clients, and last but not least communications with LDAP are described.

Key words

TCP/IP, Instant Messaging, Jabber, VoIP, SIP, LDAP, server, client, wiretap, phone system.

ŠTĚRBA, R. *Nebezpečí internetové komunikace.* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 50 s. Vedoucí bakalářské práce Ing. Michal Polívka.

Prohlášení

Prohlašuji, že svoji bakalářskou práci na téma Nebezpečí internetové komunikace jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne ....

podpis autora

Poděkování

Rád bych poděkoval vedoucímu mé bakalářské práce Ing. Michalu Polívkovi za odbornou pomoc a vedení při zpracování mé bakalářské práce.

V Brně dne .....

.....
podpis autora

Obsah

Seznam obrázků

1 Úvod

Zpočátku se práce věnuje studiu architektury TCP/IP a protokolům, které jsou zde užívány.

Dále se tato práce zabývá protokoly pro rychlou výměnu zpráv tzn. Instant Messaging (IM) a problematikou hlasové komunikace po IP (Internet Protocol) sítích. Tato práce ukazuje, jak jednotlivé IM protokoly fungují a jak jsou zabezpečeny, protože při výběru IM protokolu je jeho zabezpečení podstatný faktor.

Také jsou zde popsány protokoly pro hlasovou komunikaci a softwarové ústředny.

Jako další jsou zde rozebrány adresářové služby zvláště LDAP. Protože spravovat velké množství uživatelů bez takovéto služby, je obtížné.

Praktická část se zabývá návrhem efektivního komunikačního systému, popisem instalací jednotlivých programů (serverů) jejich dostupností, omezením a zvláště nastavením, které není leckdy triviální. A v neposlední řadě se v práci objeví propojení jednotlivých programů (např. navázání příslušných klientů).

Komunikace by měla být zefektivněna, protože je lepší nechat správu účtů specializovaným programům, než v každém programu mít vlastní účty.

2 TCP/IP

V současnosti je většina sítí založena na modelu TCP/IP. Název vznikl z nejpoužívanějších protokolů této sady – jsou to protokoly TCP a IP [1].

Zpočátku byla architektura TCP/IP použita jen pro spojení vládních počítačů v USA (síť ARPANET – předchůdce dnešního internetu). Nyní se nejvíce využívá v síti internet, jejž se stala největší celosvětovou sítí pro sdílení dat a komunikaci. TCP/IP se jako standard (skupina síťových protokolů) začal prosazovat v době, kdy byl implementován do systému UNIX a jemu podobných (80. léta). Díky této podpoře a zároveň díky vyplývající historické kompatibilitě s velkým množstvím hardwarových a softwarových systémů se dnes těší velké oblibě [2].

Při vývoji byl kladen důraz na spolehlivost a hlavně na odolnost proti výpadkům. Bohužel bezpečnost komunikace nebyla při vývoji stavěna na první místo. Tato sada protokolů je neustále rozšiřována a vylepšována, ale většina zabezpečení stále zůstává na aplikacích viz [1].

2.1 IP (Internet Protocol)

IP je protokol síťové vrstvy. Slouží k přenosu dat mezi dvěma vzdálenými počítači často přes velký počet sítí používajících různé komunikační protokoly (TCP/IP, IPX/SPX, NetBEUI, …), takovýmto příkladem je internet. Základní přenosovou jednotkou je IP-datagram. Přenos IP-datagramu je nezávislý na přenosu jiného IP-datagramu.

<table>
<tr><td colspan="4">0 8 16 24 31</td></tr>
<tr><td>Verze IP
4 bity</td><td>Délka záhlaví</td><td>Typ služby
8bitů</td><td>Celková délka IP-datagramu
16 bitů</td></tr>
<tr><td colspan="3">Identifikace IP-datagramu
16 bitů</td><td>Příznaky (flags)
Posunutí fragmentu od počátku (fragment offset)
13 bitů</td></tr>
<tr><td colspan="2">Doba života datagramu (TTL)
8 bitů</td><td>Protokol vyšší vrstvy(protocol)
8 bitů</td><td>Kontrolní součet z IP-záhlaví (checksum)
16 bitů</td></tr>
<tr><td colspan="4">IP-adresa odesílatele (source IP-address)
32 bitů</td></tr>
<tr><td colspan="4">IP-adresa příjemce (destination IP-address)
32 bitů</td></tr>
<tr><td colspan="4">Volitelné položky záhlaví</td></tr>
<tr><td colspan="4">Přenášená data (Nepoviné)</td></tr>
</table>

Obr. 1 Struktura IP-datagramu ve verzi 4 (IPv4)

Protokol IP ověřuje u každého IP-datagramu jeho korektnost a stará se o adresování.

Adresa protokolu IP verze 4 (IPv4) má délku 4 bajty. Zapisuje se dekadicky např. `192.168.1.9/24` (číslo za lomítkem znamená, kolik bitů z adresy, počítáno zleva, patří do tzv. prefixu, což je maska podsítě).

Odlišnější je adresa protokolu IP verze 6 (IPv6) má 128 bitů a nezapisuje se dekadicky po bajtech jako v IPv4 (což by vedlo k nepohodlně dlouhým řetězcům), ale zapisuje se hexadecimálně vždy po dvojicích bajtů oddělených dvojtečkami. Adresa IPv6 může vypadat následovně: `3ffe:ffff:1::baf/64`. Pokud se v adrese vyskytnou dvě dvojtečky za sebou, znamená to, že dané dva bajty mají hodnotu nula (používá se jen na páteřních sítích, současné operační systémy (OS) jej mají v sobě implementován např. OS Microsoft Windows Vista).

Adresa v protokolu IP jednoznačně identifikuje jednotlivá zařízení, či jejich rozhraní.

Protokol IPv4 neřeší bezpečnost přenášených dat. Použití některých volitelných položek záhlaví je mnohdy nebezpečné. Použití například explicitního směřování (loose source routing) může znamenat bezpečnostní riziko. Tato položka v záhlaví umožňuje zadat, přes které uzly bude IP-datagram posílán. Při takovémto směrování se mění IP-adresa příjemce, tato skutečnost umožňuje útočníkovi odklonit tok dat na jiný uzel, kde s nimi může být manipulováno. V praxi se tyto položky však v zásadě nevyužívají. Poskytovatelé internetu IP-datagramy s bezpečnostně závadnými volitelnými položkami záhlaví jako je třeba dříve zmíněné explicitní směrování nebo striktní explicitní směrování (strict source routing) zahazují [3].

Pro transport IP-datagramu sítí je důležitá jen adresa příjemce. Když směrovač (router) obdrží od jiného uzlu sítě IP-datagram, okamžitě jej pošle danému adresátovi a přitom se nedívá, od koho je tento IP-datagram posílán. Směrovač může takovouto akci provést, protože ví, na jakém portu má připojen daný uzel.

Někdy se stane, že přenášený IP-datagram je moc velký na to, aby síťové prostředky byly schopné jej přenést. Důsledkem je, že zmíněný IP-datagram musí být fragmentován (rozdělen na více menších částí). Pro tento účel poslouží položky ze záhlaví `identifikace datagramu`, `příznaky` a `posunutí fragmentu`. Tzv. defragmentaci může provést až koncový uzel.

2.1.1 ICMP

Protokol ICMP je součástí protokolu IP. ICMP slouží k signalizaci mimořádných událostí. Tento protokol balí svoje pakety do IP-datagramů (tzn. že v přenášených datagramech najdeme za linkovým záhlavím IP-protokolu záhlaví ICMP paketu).

Protokolem ICMP lze signalizovat velkou škálu situací. Konkrétní implementace TCP/IP zahrnují jen část těchto signalizací. Z bezpečnostních důvodů jsou mnohdy tyto signalizace směrovačem zahazovány.

2.1.2 Bezpečnost

Odposlech IP-datagramů je jednoduchý. Pro sledování IP-datagramů lze použít jak softwarové (Wireshark, Ethereal, ClearSight, TCP Dump, …) tak i hardwarové nástroje.

Modifikace obsahu jednotlivých IP-datagramů je snadná. Součástí modifikace obsahu IP-datagramu musí být také přepočet kontrolního součtu. V IP-datagramu je kontrolním součtem chráněna jen hlavička, datová část však nikoli.

Na úrovni IP protokolu jsou mimo jiné známy útoky zahlcení a ping smrti [4].

Některé z těchto aspektů řeší skupina protokolů IPsec.

2.2 IPsec

IPsec řeší bezpečnostní aspekty už na úrovni IP. Zabezpečení, jenž IPsec přináší, se děje na úrovni OS (mezi počítači) – IPsec nezabezpečuje data mezi uživateli na stejném počítači. Někdy zabezpečení neprobíhá mezi počítači navzájem, ale probíhá mezi hraničními uzly (server) sítě. Z toho vyplívá, že počítače v síti ani nemusí podporovat IPsec (vše si zabezpečuje server sám) a další výhoda je, že takto lze použít internet pouze jako přenosové médium.

Transportní režim IPsec používá bezpečnostní hlavičku (určuje, jak jsou data zabezpečena), kterou vloží mezi hlavičku IP-datagramu a hlavičku protokolu vyšší vrstvy.

Tzv. tunel vezme IP-datagram a „vnoří“ jej do dalšího IP-datagramu, který funguje jako obálka. K této obálce se jako v předchozím režimu přidá taktéž bezpečnostní záhlaví. Tunel má výhodu v tom, že uživatelé, kteří jsou od sebe relativně daleko (řádově i tisíce kilometrů), mohou komunikovat, jako by byli členy jedné LAN (Local Area Network).

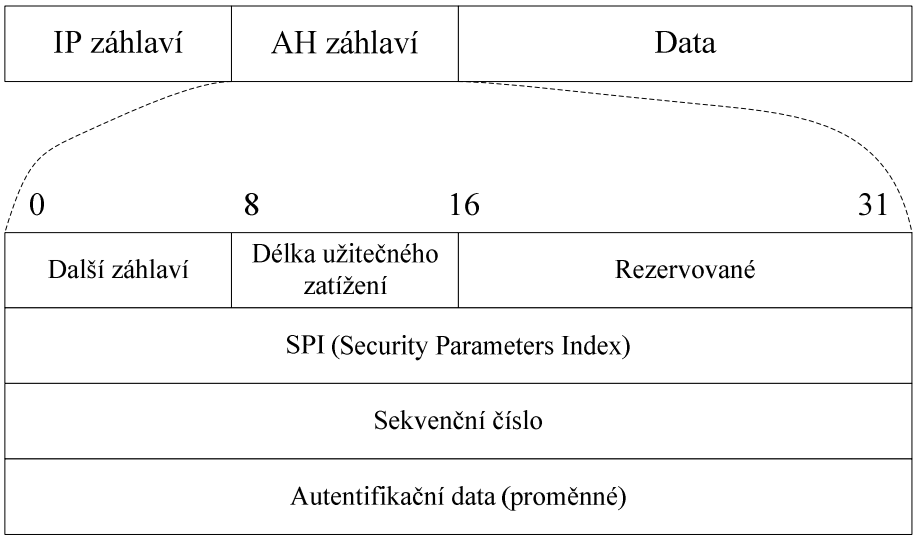
2.2.1 Bezpečnostní záhlaví

K tvorbě bezpečnostního záhlaví se používají dva režimy AH (IP Authentication Header) a ESP (IP Encapsulating Security Payload).

AH zajišťuje integritu IP-datagramů, ochranu proti útoku zopakování a autentizaci odesílatele. ESP zabezpečuje to samé a také šifrování dat.

Protokol IP poskytuje datagramovou službu a posílat informace o zabezpečení v každém IP-datagramu by nebylo moudré, protože by se zmenšil prostor pro data. Proto každé záhlaví (vytvořené pomocí AH nebo ESP) obsahuje pole SPI (Security Parameters Index). SPI je ukazatelem do databáze pro zabezpečení jednotlivých IP-datagramů. Na každém počítači jsou takováto pravidla vytvořena (nakonfigurována administrátory počítačů). Tato pravidla se označují SP (Secure Policy) a jejich souhrn SPD (Secure Policy Database).

Bohužel SPI není jednoznačný parametr pro zabezpečení. Proto se používá tzv. SA (Secute Association). SA zahrnuje index SPI, IP adresu příjemce a režim, kterým bylo vytvořeno záhlaví. Když příjemce dostane takovýto IP-datagram, tak si podle trojice SPI, adresa a režim najde v tabulce potřebné informace a dešifruje přijatý IP-datagram.



Obr. 2 Vnořené AH záhlaví

2.2.2 Databáze SA

Databáze SA se může plnit buď staticky, nebo dynamicky. Statické řešení by bylo zdlouhavé a neefektivní. Dynamicky se databáze SA plní protokolem ISAKMP. Komunikace tímto protokolem se skládá ze dvou částí. Nejdříve si protokol vytvoří zabezpečený kanál pro vlastní komunikaci a pak teprve vytváří příslušné SA. Více podrobností o této problematice lze najít v [1].

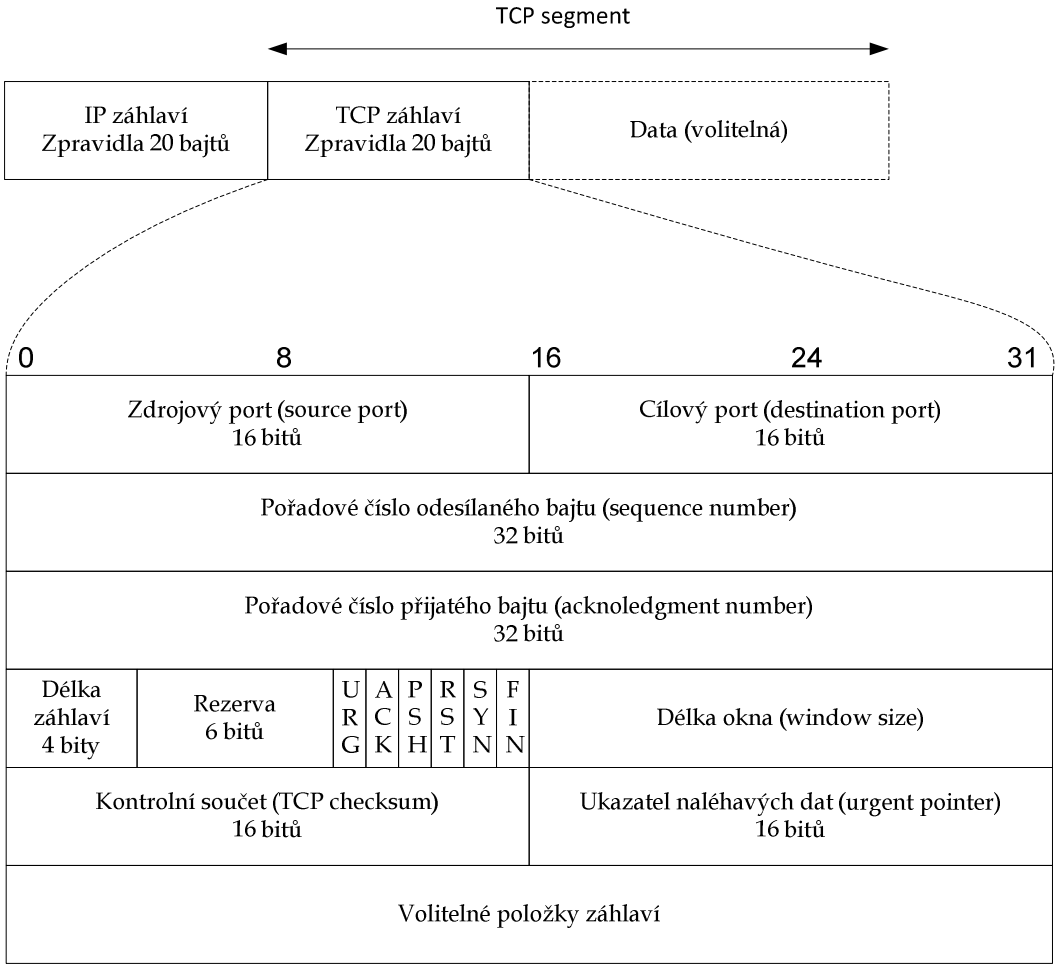
2.3 TCP (Transmission Control Protocol)

TCP je protokol transportní vrstvy. Základní přenosovou jednotkou je tzv. TCP segment. Protokol TCP přepravuje data mezi dvěma aplikacemi, zatímco IP přepravuje data mezi dvěma počítači.

Protokol TCP poskytuje aplikacím spojovanou spolehlivou službu.

Spojovaná služba (connection oriented) je služba, která mezi dvěma aplikacemi naváže spojení, tj. vytvoří na dobu spojení tzv. virtuální okruh. Pro navázání a ukončení spojení slouží příznaky (SYN, FIN a RST), které se nacházejí v záhlaví TCP segmentu. Tento okruh je plně duplexní (data se přenáší oběma směry současně).

Pro každý směr spojení čísluje přenášené bajty a sleduje, jestli nebyl tok přenesených dat přerušen. Přenesená data potvrzuje. Segmenty nesoucí potvrzení mají nastaven příznak ACK (acknowledgement = potvrzení). Nesou jej všechny segmenty kromě prvního, který navazuje spojení. Pokud zprávy dojdou nečitelné, mimo pořadí či jinak poškozené, opakuje se vysílání požadované zprávy. A proto TCP poskytuje spolehlivou službu, při ztrátě dat se opakuje vysílání.



Obr. 3 TCP segment

Číslování přenášených dat začíná po navázání spojení od náhodného čísla. Segment, který nese takto náhodně vygenerované číslo, má nastaven příznak SYN (první segment nemá nastaven příznak ACK). Takto se vygeneruje číslo pro každý směr přenosu zvlášť. Integrita přenášených dat je zabezpečena kontrolním součtem.

Službu na obou stranách určuje tzv. číslo portu. Toto číslo je dvojbajtové (nabývá hodnot 0-65535). Porty se přidělují „na pevno“, tzn. jedné aplikaci se přidělí číslo portu jednou provždy a takové přidělení se zveřejní. Pak všechny aplikační procesy mají k dispozici informace o příslušných portech na všech uzlových počítačích – těmto portům se pak také říká dobře známé porty (z anglického well-known ports). Naneštěstí tato varianta se dá použít jen pro zveřejnění takových služeb, s jejichž existencí lze předem počítat.

Další možnost je přiřazování portů aplikacím (porty se nepřiřadí na pevno). Tato varianta zvyšuje režii, ale je pružnější, protože porty jsou využívány, jak to uživateli vyhovuje (v paměti příslušného počítače nemusí být uloženy porty, které on sám nepoužívá). Avšak pro tuto možnost je zapotřebí jeden známý port, pro doptávání na použité porty.

U portů se často určuje, jakým protokol transportní vrstvy se využívá pro přenos. Musíme však podotknout, že pro protokol TCP existuje jiná sada než pro protokol UDP. Z toho vyplívá, že port se stejným číslem a jiným protokolem jsou dvě rozdílné věci (př.: 564/TCP nemá nic společného s 564/UDP).

Takže cílová aplikace je jednoznačně dána IP-adresou, portem a protokolem transportní vrstvy.

Jak u IP protokolu tak i u TCP je možná fragmentace datových jednotek. TCP fragment se vkládá do IP-datagramu a ten se vkládá do linkového rámce. Naneštěstí fragmentace zvyšuje režii, takže je lepší vytvářet TCP segmenty takové velikosti, aby fragmentace nebyla zapotřebí.

2.3.1 Bezpečnost

Jak je vidět z obr. 3 v tomto protokolu nejsou žádná bezpečnostní opatření (v TCP segmentu není žádné pole, které zabezpečuje obranu dat proti zcizení). Avšak jedna věc by se tak mohla brát, je to náhodné zvolení čísla pro začátek komunikace. Například když komunikuje klient se serverem a po chvíli bude komunikace přerušena, pak se komunikace znova naváže (předpokládejme, že nová komunikace probíhá na stejném portu jako předešlá), následně se objeví zpožděné segmenty z předešlého spojení (mohou být upraveny útočníkem), ale protože nové spojení začínalo od nového čísla, tak se zpožděné segmenty zahodí.

Dnes jsou také běžně dostupné programy, díky nimž může útočník převzít spojení za klienta. To se může stát, pokud komunikace není dostatečně zabezpečena (např. zabezpečení krátkým klíčem).

Pak je tu známé zahlcení. Tentokrát nejde o zahlcení sítě, ale o vyčerpání zdrojů. TCP je spojově orientován a tudíž si server musí pamatovat informace o aktuálních spojeních. Útočníkovi jde o to, aby navázal velký počet spojení se serverem a vyčerpal jeho systémové zdroje. Jde o tzv. útok DoS (Denial of Service).

Útočník může také vyhledávat porty, na kterých server naslouchá (tj. očekává požadavky klientů). Podle portů pak může usoudit typ aplikace a následně útočit na ni. Ale aby zjistil, na jakých portech server naslouchá, musí se s nimi pokusit navázat spojení. Nejjednodušší je začít od portu s číslem 0 a pokračovat dál. Moderní servery resp. firewally takové scanování umějí rozpoznat. Takže útočníkovi nezbude nic jiného než porty testovat náhodně a roztáhnout testování na delší časové období.

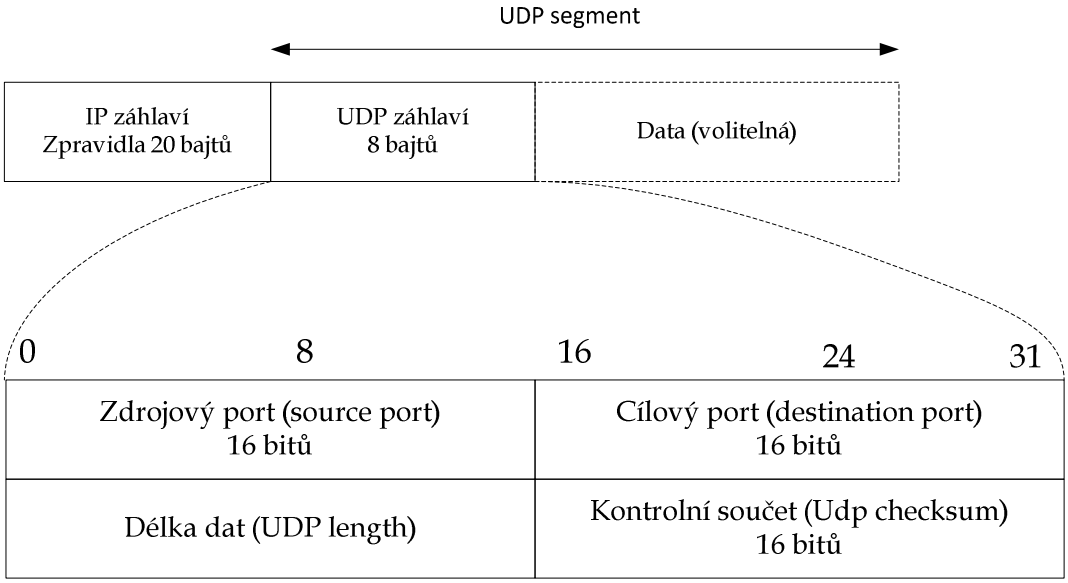
2.4 UDP (User Datagram Protocol)

UDP je také protokol transportní vrstvy jako TCP, ale na rozdíl od něj UDP poskytuje nespojovanou nespolehlivou službu. Základní transportní jednotkou je UDP datagram.

Nespojovaná služba znamená, že se nevytváří spojení. Jednoduše se posílají UDP datagramy s adresou cíle a dál už záleží na síti, jakou cestou přijde k příjemci.

Protokol UDP je vhodný pro aplikace, které nepožadují trvalé spojení nebo potvrzení doručených dat (aplikace pro realtime – volání přes intenet).

Na protokolu UDP jsou také založeny všechny aplikace, které používají na svoje šíření adresní oběžníky (multicast).



Obr. 4 UDP datagram

3 Protokoly pro Instant Messaging a jejich odposlech

Instant Messaging (IM) je služba umožňující textovou komunikaci dvou a více uživatelů v reálném čase. Používají se na to různé protokoly a různí klienti (komunikace server-klient).

Odposlech je často využíván správci sítě, když hledají nějakou chybu. Mohou použít jak softwarové analyzátory, tak i hardwarové. Problém nastává, pokud se odposlech provádí se zlým úmyslem (ukrást nějaká data). Aby takováto situace nemohla nastat, je potřeba zabezpečit komunikaci. Dále uvidíme, jak jsou na tom se zabezpečením různé IM protokoly.

3.1 Protokol ICQ

V České republice je asi nejrozšířenější protokol právě ICQ (=I seek you neboli česky hledám tě). Dnes je protokol ICQ vlastnictvím firmy AOL. Tato firma jej koupila od Izraelské firmy Mirabilis, která jej vyvinula.

ICQ operuje s tzv. UIN (Universal Internet Number), je to něco jako telefonní číslo. Tento UIN přidělí uživateli systém při registraci. Domovský server ICQ je www.icq.com, na kterém se nachází centrální databáze.

3.1.1 Jak ICQ funguje

Přihlášení probíhá následovně. Po zadání UIN a hesla klient kontaktuje ICQ-server kvůli přihlášení (login.icq.com) a dostane odpověď (pokud server není zaneprázdněn). Tato komunikace probíhá přes protokol UDP.

Teď se může klient spojit se serverem protokolem TCP a vytvořit virtuální okruh. Toto spojení se vytvoří klasicky, jak je to obvyklé na transportní vrstvě: klient pošle paket s příznakem SYN (zahajuje spojení), server mu na to odpoví paketem s příznaky SYN a ACK (potvrzení a zahájení spojení) a pak klient pošle jen ACK. Teď, když klient ví, že server naslouchá, tak klient posílá svoje UIN a heslo (UIN není zašifrované, takže útočník okamžitě zjistí, jaká osoba se přihlašuje, naštěstí heslo zabezpečeno je – použit je hash anebo MD-5), také si domluví parametry spojení (jakou verzi protokolu podporuje klient; jaký jazyk je nastaven např. angličtina; …). Po domluvení těchto informací je stávající spojení ukončeno (použijí se pakety s příznakem FIN).

Následně se tato komunikace opakuje, je úplně stejná, ale na závěr spojení server pošle klientovi tzv. autorizační cookie, kterým se pak přihlásí (délka cookie je 256 bajtů, což není špatné, ale takhle někdo může toto cookie odposlechnout a převzít komunikaci za klienta).

Server po ukončení spojení okamžitě iniciuje vznik dalšího spojení a vyžaduje od klienta tzv. autorizační cookie. Po výměně několika dalších informací začíná server posílat klientovi jeho seznam kontaktů, posílá se v několika TCP segmentech. Jeden záznam v seznamu kontaktů odpovídá jednomu UIN a NICK (přezdívka uživatele, jemuž náleží příslušný UIN). Oba dva parametry nejsou opět šifrovány.

Od serveru dostaneme kontaktní informace právě přihlášeného uživatele (to může být velké množství informací např. bydliště e-mailová adresa, číslo telefonu a další). Server průběžně informuje o aktuálních statusech jiných uživatelů, které máme v seznamu kontaktů (server posílá dané UIN a status ve stringové hodnotě – text statusu). Také téměř okamžitě posílá zprávy o změnách stavu uživatelů (jestli se někdo ze seznamu kontaktů nepřihlásil nebo neodhlásil). Žádná z informací zmíněná v tomto odstavci opět není nikterak zabezpečena.

```
⊟ AOL Instant Messenger
    Command Start: 0x2a
    Channel ID: New Connection (0x01)
    Sequence Number: 38576
    Data Field Length: 91
    Protocol Version: 00000001
  ⊟ TLV: Screen name
      Value ID: Screen name (0x0001)
      Length: 9
      Value: 279073730
  ⊟ TLV: Roasted password array
      Value ID: Roasted password array (0x0002)
      Length: 8
      Value
  ⊟ TLV: Client id string (name, version)
      Value ID: Client id string (name, version) (0x0003)
      Length: 8
```

```
0000  00 17 a4 c2 09 00 00 0e  2e 64 07 8b 08 00 45 00   ........ .d....E.
0010  00 89 01 46 40 00 80 06  ef 13 93 e5 94 6a 40 0c   ...F@... .....j@.
0020  a1 b9 04 17 14 46 0c 8f  12 f7 a4 3c 23 2d 50 18   .....F.. ...<#-P.
0030  45 06 d8 68 00 00 2a 01  96 b0 00 5b 00 00 00 01   E..h..*. ...[....
0040  00 01 00 09 32 37 39 30  37 33 37 33 30 00 02 00   ....2790 73730...
0050  08 c7 15 b0 fc 01 b2 e8  a3 00 03 00 08 49 43 51   ........ .....ICQ
0060  42 61 73 69 63 00 16 00  02 01 0a 00 17 00 02 00   Basic... ........
0070  14 00 18 00 02 00 34 00  19 00 02 00 00 00 1a 00   ......4. ........
0080  02 0b b8 00 14 00 04 00  00 04 3d 00 0f 00 02 65   ........ ..=....e
0090  6e 00 0e 00 02 75 73                              n....us
```

Obr. 5 Odposlech segmentu nesoucí UIN a heslo

```
          Client Capabilities Flags: 0x00000003
          Unknown
          Downcounter?
          Length: 14
          Downcounter?
          Unknown
          Message Type: Plain text (simple) message (0x01)
        ⊞ Message Flags: 0x00
          Status Code: 0
          Priority Code: 4
          Text Length: 44
          Text: Ahojky, jak se máš? dneska je venku hezky
  ⊞ TLV: Unknown
```

```
0020  94 6a 14 46 04 18 99 c2  9e 22 88 58 ef ae 50 18   .j.F.... ."·X..P.
0030  40 00 b3 3a 00 00 2a 02  4a c0 01 24 00 04 00 07   @..:..*. J..$....
0040  00 00 c7 f4 60 81 00 00  f0 6f bd b7 00 00 00 02   ....`... .o......
0050  09 33 36 38 39 32 34 31  38 30 00 00 00 06 00 01   .3689241 80......
0060  00 02 00 50 00 06 00 04  10 01 00 00 00 05 00 04   ...P.... ........
0070  45 d5 61 7c 00 1d 00 14  00 01 01 10 36 04 22 a1   E.a|.... ....6."·
0080  a1 1e 51 89 86 34 75 39  f4 ff 38 16 00 0f 00 04   ··Q··4u9 ··8.....
0090  00 00 0f 3f 00 03 00 04  49 0a c1 c2 00 05 00 bb   ...?.... I.......
00a0  00 00 00 00 f0 6f bd b7  00 00 09 46 13 49 4c 7f   .....o.. ...F.IL.
00b0  11 d1 82 22 44 45 53 54  00 00 00 0a 00 02 00 01   ...·"DEST ........
00c0  00 0f 00 00 27 11 00 93  1b 00 0a 00 00 00 00 00   ....'... ........
00d0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 03 00   ........ ........
00e0  00 00 00 70 31 0e 00 70  31 00 00 00 00 00 00 00   ...p1..p 1.......
00f0  00 00 00 00 00 01 00 00  00 04 00 2c 00 41 68 6f   ........ ...,.Aho
0100  6a 6b 79 2c 20 6a 61 6b  20 73 65 20 6d c3 a1 c5   jky, jak  se m···
0110  a1 3f 20 64 6e 65 73 6b  61 20 6a 65 20 76 65 6e   ·? dnesk a je ven
0120  6b 75 20 68 65 7a 6b 79  00 00 00 00 00 ff ff ff   ku hezky .·····
0130  00 26 00 00 00 7b 30 39  34 36 31 33 34 45 2d 34   .&···{09 46134E-4
0140  43 37 46 2d 31 31 44 31  2d 38 32 32 32 2d 34 34   C7F-11D1 -8222-44
0150  34 35 35 33 35 34 30 30  30 30 7d 00 13 00 01 c9   45535400 00}·····
```

Obr. 6 Odposlech zprávy

Před vlastním přenosem zprávy dostane klient tzv. upozornění MTN (Mini Typing Notification), které nás informuje o skutečnosti, že nám někdo píše zprávu. Abychom věděli, od koho vlastně nám zpráva přijde, tak v MTN je obsažen UIN tohoto uživatele.

Přenos textové zprávy v protokolu ICQ je zase nešifrován (jak je vidět na obr. 6). Takže pro útočníka není problém si přečíst jakoukoli zprávu. K textové zprávě se zpravidla připojují další informace jako třeba UIN a status odesílatele popřípadě i text přidružený k statusu. Jako obvykle ani jedna z těchto věcí není zašifrována.

Protokol ICQ je příjemný pro uživatele a kromě IM komunikace podporuje spoustu dalších věcí (např. online hry). Na druhou stranu protokol ICQ nezabezpečuje přenos dat (resp. komunikaci). Takže uživatelé, kteří nechtějí, aby jejich komunikaci někdo odposlechnul a následně zjistil citlivé údaje, by měli použít bezpečnější protokol.

3.2 Protokol MSN

Tento protokol pochází z dílny firmy Microsoft. Odtud je také název `The MicroSoft Network` neboli MSN. Tento protokol v ČR moc rozšířen není, jeho doménou je především Severní a Jižní Amerika. První verze vznikla v roce 1999 a uměla pouze posílání zpráv a tvoření seznamu kontaktů. Dnešní verze už jsou daleko vyspělejší.

3.2.1 Jak se přihlásit do MSN

Nejdříve se musí vyplnit kontaktní informace. Uživatel se bude přihlašovat e-mailovou adresou, ta má zaručenou jedinečnost na internetu. Dále se musí registrovat tato adresa na serveru, jehož vlastníkem je Microsoft a poté potvrdit, že zadaná adresa je opravdu uživatele, jenž ji zadal.

Průběh přihlášení je následující. Nejdříve se naváže virtuální okruh (protokol TCP) standardně segmenty s příznaky SYN a ACK. Server a klient se dohodnou, jakou verzi protokolu budou používat. Dále klient posílá svoji verzi (programu, jež používá) a e-mailovou adresu přihlašujícího (ta zatím není šifrována). Klient obdrží socket (IP adresa a port) nového serveru, přes který posléze bude prováděna IM komunikace (server, se kterým komunikuje teď, slouží jen pro výměnu dat nezbytných pro další komunikaci).

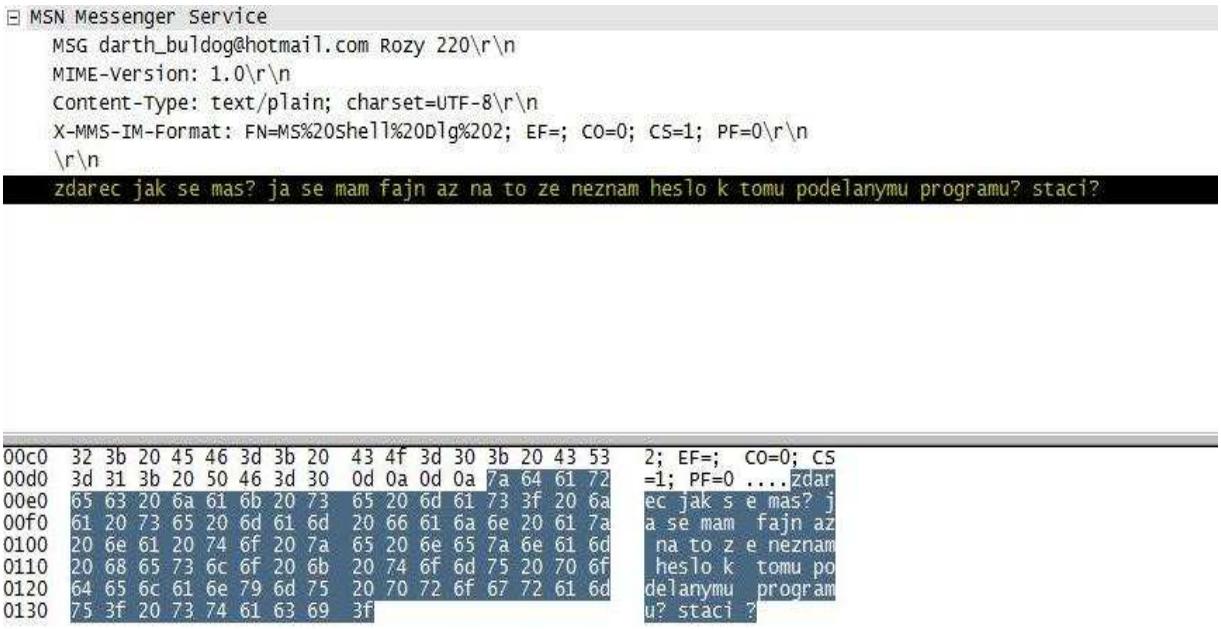
Následně se naváže spojení s novým serverem na portu, jenž byl přidělen. Celá autentizace (ověření uživatele, tj. zdali souhlasí uživatelské jméno a heslo) probíhá přes Secure Socket Layer neboli SSL. Takže konečně je jen autentizace zašifrována. Nejdříve server prokáže svou identitu pomocí certifikátu (dochází k tomu v tzv. zprávě Server Hello), následně si vymění šifrovací klíče a další algoritmy pro určení kontrolního součtu (sever přikáže klientovi, aby si změnil tzv. šifrovací provedení). Pak teprve přichází toužená výměna dat, v našem případě autentizace.

3.2.2 Posílání zpráv pomocí protokolu MSN

Pro posílání textových zpráv používá MSN normu MIME (Multipurpose Internet Mail Extension). Bohužel MSN používá pouze hlavičky MIME-Version a Content-Type (specifikuje pouze typ dat, která jsou přenášena), takže vlastní zprávy nejsou zašifrovány. Jak lze vidět na obr. 7 odposlech takových zpráv je jednoduchý.

3.2.3 Závěr

Výrazný pokrok v zabezpečení oproti ICQ tady existuje. Přenos uživatelského jména (e-mailová adresa) a hesla je zabezpečeno pomocí SSL. Zprávy sice nejsou nikterak zabezpečeny proti odposlechu, ale MSN využívá normu MIME, která specifikuje šifrování.



Obr. 7 Odposlech zprávy MSN

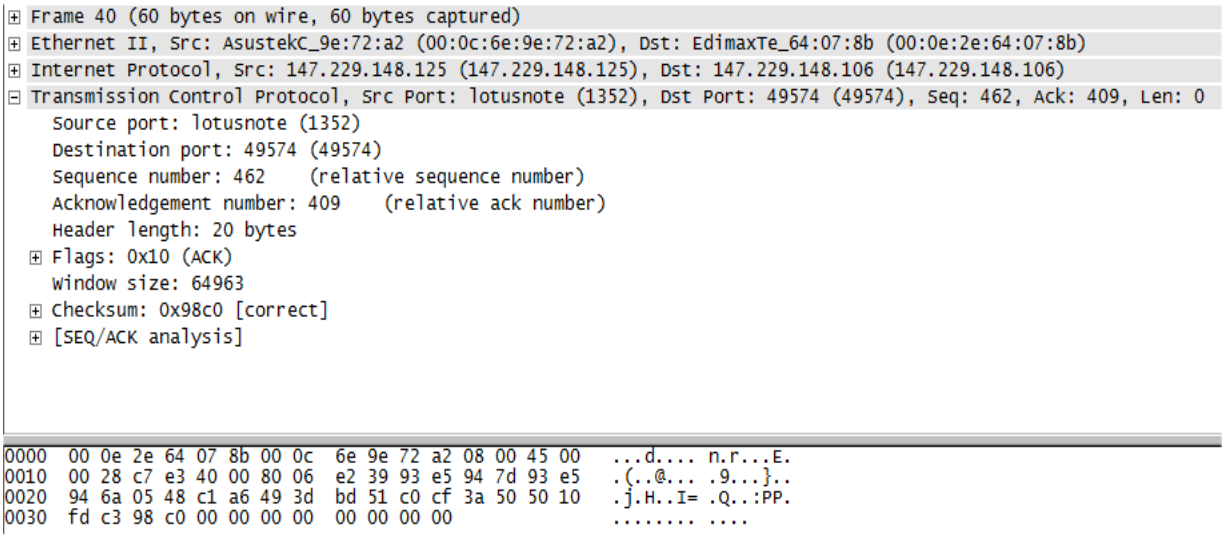
3.3 Skype

Skype je rychle se rozšiřující komunikační protokol (stejný název má i program), který si získává příznivce po celém světě. Tento protokol umožňuje peer-to-peer komunikaci (komunikace probíhá mezi klienty navzájem, server jen kontroluje zabezpečení). Pod toto dílo se mohou podepsat Niklas Zennström a Janus Friis, kteří již dříve vytvořili dobře známý software Kazaa (program na peer-to-peer sdílení dat). Skype poskytuje IM a internetovou telefonii (VoIP).

3.3.1 Komunikace přes Skype

Při registraci se nepřiděluje žádné UIN nebo nezadává e-mailová adresa, ale tentokrát si uživatel zvolí svoje vlastní tzv. Skype jméno, které bude jedinečné jen mezi uživateli Skype. Dále se zadá heslo a jako obvykle si uživatel může vyplnit kontaktní informace.

Skype má výborně řešenou ochranu probíhající komunikace, jedná se o silnou šifru 2048 bitové RSA (šifrování s použitím veřejného klíče) pro výměnu klíče a pro šifrování přenosu přenosu 256 bitové šifrování AES (Advanced Encryption Standard) [5]. To dává protokolu Skype jednoznačnou výhodu mezi ostatními IM protokoly na poli zabezpečení proti odposlechu. Data nesená TCP segmenty nebo UDP datagramy nemají pro útočníka žádný význam, neboť jsou zašifrována. Tato skutečnost je názorně ukázána na obr. 8.



Obr. 8 Odposlech komunikace přes Skype

3.3.2 Klady a zápory

Skype je chráněn RSA a AES, z toho plyne, že jakýkoli odposlech od útočníka je celkem nemožný. Takže uživatelé používající Skype nemusí vůbec trápit obava, jestli jejich konverzace poslouchá nějaký druh útočníka [6].

Samotný protokol Skype je uzavřený. Firma vlastnící Skype říká uživatelů souhlasící s licenční smlouvou, že v případě soudního rozhodnutí je možné, že jejich rozhovory budou odposlouchávány. Server může dát klientovi příkaz, aby nešifroval vůbec nebo aby bylo šifrování oslabeno. Pak už nařídí klientovi, aby komunikaci směřoval přes jistý uzel, kde může firma odposlouchávat. Takže uživatel nikdy neví, jestli není jeho komunikace odposlouchávána ze strany poskytovatelů [6].

Uživatel disponující veřejnou IP-adresou je protokolem Skype využíván jako ústředna a mohou přes něj téci komunikační data a audio proudy. Z toho vyplývá, že uživatel nemá vůbec žádnou kontrolu nad přenesenými daty (cestu pro data vybírá server). Tato skutečnost by se už dala považovat za útok, když někdo využívá váš hardware a připojení k internetu bez vašeho souhlasu. Skype toto využívá proto, aby mohl spojit dva uživatelé bez veřejné IP-adresy (tito uživatelé mohou být za striktními firewally) [6].

Občas dojde také k nemalému zpoždění, protože routing (směrování) není úplně ideální. Třeba když uživatelé komunikující na krátkou vzdálenost (řekněme desítky a stovky kilometrů) mohou být jejich komunikace směrována i přes jiný kontinent [6].

Celkově vzato je Skype velice dobře zabezpečen proti vnějším útokům. Má však i své stinné stránky, které již byly popsány výše. Takže záleží pouze na uživateli, jestli bude využívat Skype ke své komunikaci.

3.4 Jabber

Protokol Jabber neboli XMPP (Extensible Messaging and Presence Protocol) je založen na XML (eXtensible Markup Language), což je značkovací jazyk podobný HTML (HyperText Markup Language) [7]. Jabber je decentralizovaný, tzn. všechny servery jsou na sobě nezávislé. To je výhoda, protože když spadne jeden server, tak ostatní uživatele komunikují dál. Každý si také může na svém počítači nainstalovat server Jabber [8].

Jabber používá pro identifikaci uživatele adresu (JID neboli Jabber IDentification), jež má následující tvar `jmeno@domenaserveru.idf` (idf je identifikace země, např. cz). Takže JID nese informaci o uživateli i o serveru, kam se přihlašuje.

3.4.1 Komunikace

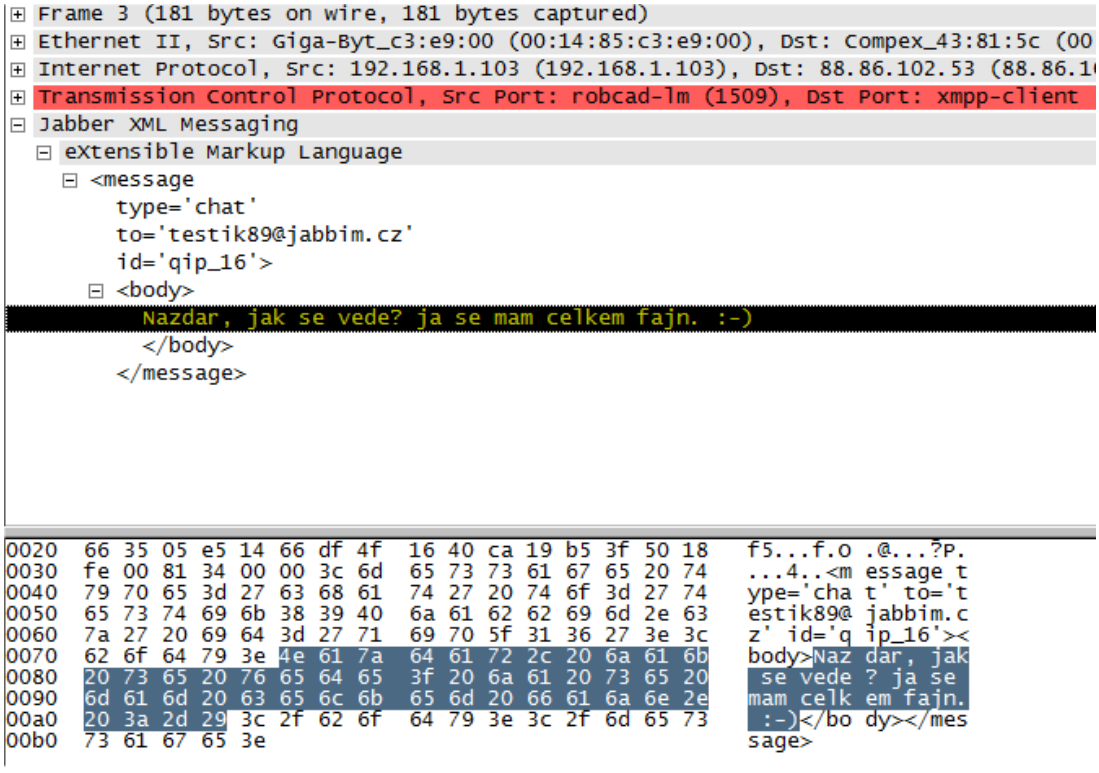
Nejdříve se klient dotáže pomocí DNS (Domain Name System), jestli spojení může být navázáno. Poté server odpoví a sdělí mu, kam se má připojit pro další komunikaci. Pak je spojení navázáno.

Klient posílá serveru komunikační parametry, které využívá (např. verze XML nebo způsob šifrování). Server odpoví ACK také odešle svoje parametry. Dále přijde dohodnutí parametrů pro autentizaci (jaká se bude používat komprese např. zlib a šifrování), tuto výměnu iniciuje server. Následuje zabezpečená autentizace (většinou se využívá MD-5).

Znovu se zopakuje výměna komunikačních parametrů. Server se také dozví, jakého klienta uživatel používá např. QIP Infium (Quiet Instant Pager). Server pošle klientovi adresy, na kterých se nachází popisy ke standardům, podle kterých se řídí komunikace.

Dále se pošle seznam kontaktů, pro jednotlivé položky se posílají údaje jméno, JID a souhlas (jestli uživatel souhlasí se sdílením kontaktních informací). Odesílání seznamu kontaktů je nešifrované.

Odesílání zpráv probíhá následovně: klient se dotáže serveru, jestli může odesílat, server mu odpoví a klient pošle zprávu. Server potvrdí, že dostal správu. Tato zpráva je opět nešifrována.



Obr. 9 Odposlech zprávy Jabber

3.4.2 Bezpečnost

Jabber, jak již bylo zmíněno, implicitně nešifruje zprávy. To však lze změnit. Jabber může využívat SSL/TSL (Secure Socket Layer/Transport Security Layer). SSL/TSL může být implementováno přímo v klientovi nebo případně lze doinstalovat. Při používání SSL/TSL bude celá komunikace probíhat šifrovaně.

Komunikace s využitím SSL/TSL funguje následovně. Klient se spojí se serverem a dohodnou se na použití komunikačních parametrů (šifrovacích algoritmů, použitých klíčů, …). Pak si prokážou totožnost. Následuje vlastní zašifrovaná komunikace. Při použití SSL/TSL může útočník využít útok tzv. muž uprostřed (z anglických man in the middle) – útočník zdánlivě vypadá jako cílový server (může toho dosáhnout třeba podvržením DNS odpovědi) a pak místo něj přebírá a dešifruje data. Proti tomuto se lze bránit certifikací. [9]

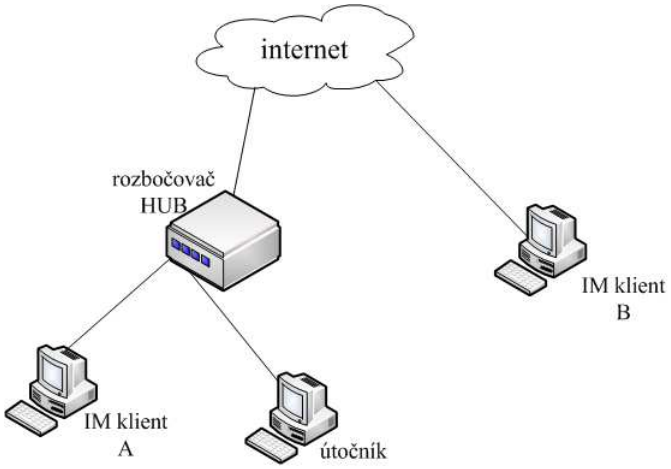
3.4.3 Závěr

Protokol Jabber šifruje autentizaci a posléze dovoluje použít SSL/TSL. Z toho vyplívá, že komunikace může být zcela šifrována. Výpadek jednoho serveru neznamená, že celá síť zkolabuje, to umožňuje decentralizace. A dále Jabber podporuje IM transport, tj. vezme jiný IM protokol a zahrne ho do sebe. Posílá zprávy, ty jdou na Jabber-server a pak se oddělí a jdou na cílový server (např. ICQ na AOL server). Z toho plyne, že takto lze šifrovat i ICQ.

Jabber je tedy z dříve zmíněných IM protokolů nejlepší volbou. A proto bude použit jako IM server Jabber server. Dostupných Jabber serverů není málo (ejabberd, jabberd2, jabberd14, …). Všechny uvedené servery pracují pod OS Linux. Ale pouze server ejabberd pracuje jak pod OS Linux tak i pod OS Microsoft Windows. Další výhoda serveru ejabberd je, že umí komunikovat přímo se serverem LDAP. Další nesporná výhoda je, že server ejabberd má pro konfiguraci a řízení intuitivní webové rozhraní. Zvláště kvůli těmto výhodám ale nejen kvůli nim byl vybrán server ejabberd jako IM server pro tuto práci.

4 Realizace odposlechu

K realizaci odposlechu byly použity tři počítače. Dva z nich byly zapojeny do rozbočovače (HUB), kvůli snazšímu odposlechu (HUB rozesílá všechna příchozí data na každý port, vyjma portu, odkud informace přišla). Tento HUB, stejně jako poslední stanice, byl následně připojen k internetu.

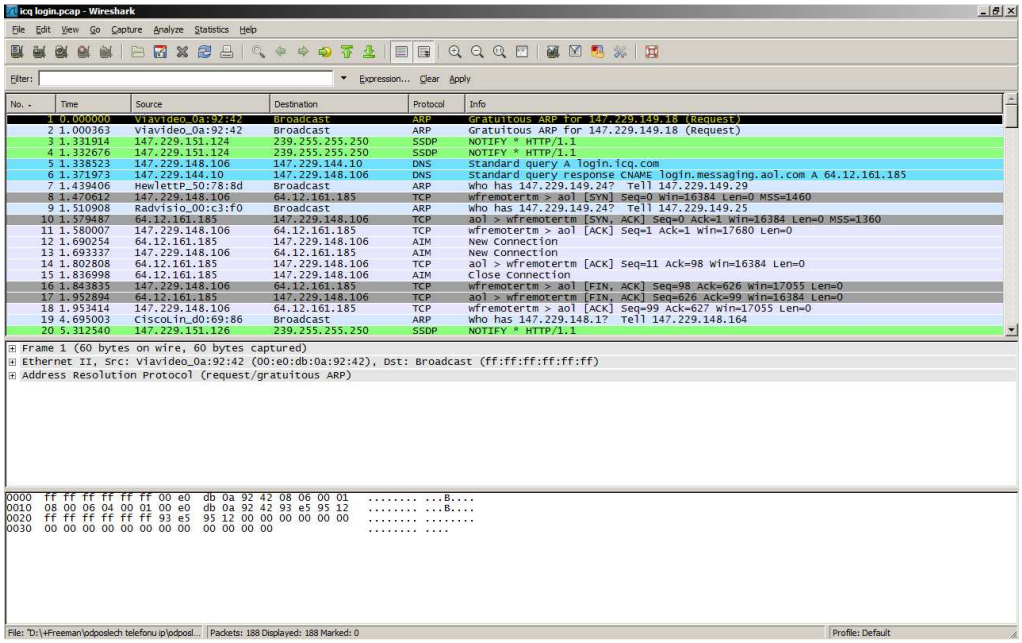


Obr. 10 Zapojení pro odposlech

Na dvou stanicích byl nainstalován příslušný IM klient (byly použity klienti QIP Infium, QIP verze 8070, Skype 3.8 a MSN Messenger 7.5) a stanice, která je označena „útočník“ byla vybavena programem na zachytávání komunikace (wireshark).

Okno programu wireshark je rozděleno na tři části. V první jsou zobrazeny příchozí segmenty (obsahuje IP adresu příjemce i odesilatele, čas zachycení apod.). V další části se nachází popis segmentu. A v poslední části jsou vypsána data příchozího segmentu.

Wireshark dovoluje i aplikaci filtrů a tím zjednodušit vyhledávání potřebných segmentů.



Obr. 11 Wireshark

Na stanici označené útočník se spustil program wireshark a následně aplikoval filter na zobrazení segmentů pouze od počítače A, který je připojen do HUBu (byl použit filter na IP adresu). Nyní může komunikovat stanice A se stanicí B a všechna jejich komunikace bude zachytávána na počítači označeném útočník.

Útočník z jejich komunikace může zjistit IM protokol, jejž používají stanice A a B. Pokud není jejich komunikace šifrována, tak útočník zachytí text zprávy. Může zjistit, jaké protokoly byly použity (TCP, UDP), také přes jaké servery dané stanice komunikují. A mnoho dalších informací.

Takto byly odposlouchávány všechny IM protokoly, jež byly zmíněny v předcházející kapitole.

5 VoIP

Technologie VoIP (Voice over Internet Protocol) umožňuje provádět telefonické hovory prostřednictvím IP sítí (např. internet nebo místní síť LAN). Hlas účastníka takového hovoru je digitalizován a převeden na pakety (buď VoIP telefonem nebo příslušným softwarem). Tyto pakety jsou pak prostřednictvím IP sítě doručeny příjemci, kde jsou převedeny zpět na hlas.

Aby mohlo být spojení kvalitní, neboli aby si účastníci využívající VoIP navzájem rozuměli (nedocházelo k velkému zpoždění nebo nadměrným výpadkům) a spojení se neukončilo předčasně, musí být zajištěna tzv. kvalita služby neboli QoS (Quality of Sevice).

Telefonní hovor je zakódován podle jistého algoritmu (pro zmenšení objemu přenášených dat). Tyto algoritmy jsou standardizovány a nazývají se kodeky (zkratka z „kódovací a dekódovací algoritmy“). Označují se např. G.711, G.723, G.729, apod.

Protokolů pro VoIP je celá řada. Nejpoužívanější jsou H.323 a SIP (Session Initiation Protocol). Některé firmy využívají také vlastní protokol pro komunikaci. Dále se také využívá protokol IAX2, jenž je protokolem ústředen Asterisk.

Pro uskutečnění VoIP hovoru postačí koncová zařízení a spojovací médium. Většinou se používá i jiná zařízení (k rozšíření služeb) např. brány (gateway), gatekeeper, MCU (Multipoint Control Unit) [10].

5.1 H.323

H.323 je primárně určen pro přenos multimediálních dat. H.323 je standard, kterému jsou podřazeny jisté protokoly např. RTP (Real-time Transport Protocol) a kodeky (G.7xx pro zvuk a H.26x pro video).

Často využívaný zvukový kodek je G.711, jenž kóduje pomocí PCM (Pulse-Code Modulation, česky pulzně kódová modulace) s maximální datovou rychlostí 64 kb/s. Stejnou kvalitu hovoru dokáže poskytnou i kodek G.722 při stejné rychlosti. Kodek G.723 potřebuje pro hovor se stejnou kvalitou pouze 8 kb/s [11].

5.1.1 Prvky H.323

Jak již bylo zmíněno výše H.323 využívá různé komponenty [10].

- Terminál – slouží k přímé komunikaci mezi uživateli. Může jím být jak IP-telefon tak příslušný software.
- Brána – je určená pro překlad protokolů. Když jeden účastník je mimo síť H.323, tj. v síti, která není postavena na H.323, tak brána zajistí, aby telefonní rozhovor mohl být vytvořen.
- Gatekeeper – je ústřední bod pro volání v síti, i když není povinou součástí H.323. Poskytuje řadu služeb např. autorizaci a autentizaci terminálů, účtování apod.
- MCU – stará se o konferenční spojení mezi terminály, tj. stará se o spojení mezi více než dvěma terminály současně.

Prvky brána, gatekeeper a MCU jsou logicky odděleny, ale v praxi se často setkáváme se skutečností, že tyto komponenty bývají seskupeny do jednoho zařízení.

5.1.2 H.323 zóna

H.323 zóna je souhrn všech terminálů, bran a MCU, které jsou řízeny pouze jedním gatekeeperem.

5.1.3 Protokoly H.323

Již bylo poukázáno, že H.323 není samostatný protokol nýbrž souhrn protokolů. A patří k nim následující [10]:

- H.225-RAS – používá se mezi koncovým bodem a gatekeeperem. Jeho účelem je registrace, zjištění stavu, kontrola přístupu apod. Spojení terminálu a gatekeeperu pomocí H.225-RAS se využívá na signalizaci a pro přenos využívá UDP.
- H.225 – přenos se uskutečňuje mezi dvěma koncovými body. A přenáší se signalizační zprávy protokolu H.225. Využívá se pro přenos TCP.
- H.245 – jedná se o řídící signalizaci, která probíhá před přenosem multimediálních dat. Vyměňují se kontrolní zprávy mezi koncovými body. Tyto zprávy se vztahují k vyjednání schopnosti výměny zpráv, kontrole toku, přenosu zpráv atd.
- T.120 – používá se pro přenos textových zpráv mezi uživateli (text není komprimovaný).
- RTP – stará se o doručení multimediálních dat (audio + video) ke koncovým bodům. Používá pro přenos UDP.
- RTCP (Real-time Transport Control Protocol) – je řídící protokol pro RTP, poskytuje informace o přenášených datech

Více informací o rodině protokolů H.323 i jiné informace o H.323 lze nalézt v [10].

5.2 SIP

SIP byl přijat jako standard v roce 1999 pod označením RFC 2543. O jeho rozvoj se stará hlavně IETF (Internet Engineering Task Force, česky Komise techniky internetu). Dnešní verze SIP nese označení RFC 3261. [12]

SIP je textově orientovaný signalizační protokol, jehož účelem je navazování komunikačních relací mezi dvěma a více koncovými body. Je hojně používán pro signalizaci sestavení, modifikaci a ukončení spojení. SIP patří mezi protokoly aplikační vrstvy a využívá povětšinou transportní protokol UDP (výjimečně i TCP). Obvykle také používá port 5060.

5.2.1 Adresace v protokolu SIP

K identifikaci UA (User Agent, koncový bod) se používá tzv. SIP URI (Uniform Resource Identifier, česky jednotný identifikátor zdroje). Toto URI obsahuje uživatelské jméno (nebo číslo) a doménu. Tyto dvě položky se oddělují znakem @. Obecný tvar vypadá takto: `sip: uživatel@doména` [10].

Příklady SIP URI:

- `sip: 103@147.229.151.16`
- `sip: michal@feec.vutbr.cz`

SIP podporuje jak webovou adresaci (SIP URI), tak i telefonní čísla (uživatelé dosažitelné prostřednictvím brány), což mu dává velkou výhodu – přechod mezi telefonní sítí a internetem. IP-telefony nebo počítače s příslušným hardwarem mohou komunikovat přímo, pokud znají URI druhé strany [12].

5.2.2 Základní prvky SIP

Dva základní prvky SIPu jsou UA a server.

UA (User Agent) představuje koncové zařízení (buď IP-telefon, příslušný software tzv. softphone nebo brána), které se spojuje s ostatními UA. UA se dále dělí na UAS (User Agent Server)a UAC (User Agent Client), které jsou obsaženy v každém koncovém zařízení. UAS reaguje na žádosti a posílá na ně příslušné odpovědi a UAC inicializuje spojení [10].

Server při používání SIP není nutný (dvě koncové zařízení mohou komunikovat mezi sebou), ale často se používá na spojení jednoho účastníka hovoru s druhým. Rozlišují se tři typy serverů.

- Proxy server – přeposílá žádosti jiných proxy serverů nebo koncových UA dalším proxy serverům nebo UA.
- Registrar server – přijímá registrace (žádosti) od UA. V databázi si ukládá soupis UA, které jsou zaregistrovány u příslušné domény.
- Redirect server – po přijetí žádosti o spojení, odpoví příslušnému zařízení, kam se má obrátit, aby se mohl spojit s cílovou stanicí.

5.2.3 Funkce SIP

Protokol SIP má několik funkcí. Patří k nim následující:

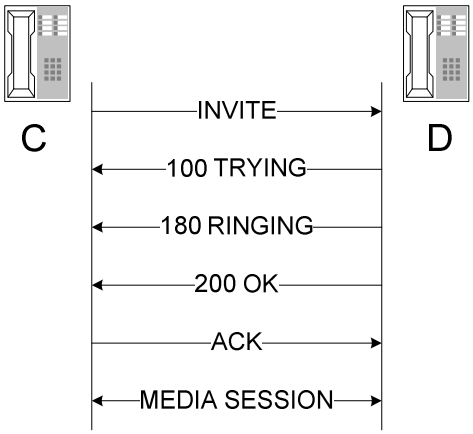
- Rozlišení adres – může být vykonáno jak UA, tak i serverem. Rozlišení se skládá z několika kroků: nejdříve se zkontroluje DNS záznam a stanovení protokolu, dále se stanoví DN (domain name, česky doménové jméno) serveru a portu, také se stanoví IP adresa hosta.
- Sestavení relace – k sestavení relace se používá zvláště zpráv INVITE, 200 a ACK. Pokud se ve spojení vyskytnou chyby, tak se požívá místo 200 4xx, 5xx nebo 6xx. Více k sestavení relace je v kapitole 5.2.4.
- Řízení hovoru – SIP je také navržen pro řízení hovoru mezi UA. Leckdy se používá řízení hovoru třetí stranou. Buď se použije tzv. hlídač, který přijímá žádosti INVITE a odpovídá na ně, dále kontroluje signalizaci a přeposílá SDP (Session Description Protocol) zprávy, nebo se použije speciální metoda REFFER – tím se inicializuje řízení třetí strany.
- Signalizace hovorů
- Nastavení QoS

Více k funkcím SIP lze nalézt v [10].

5.2.4 Komunikace SIP

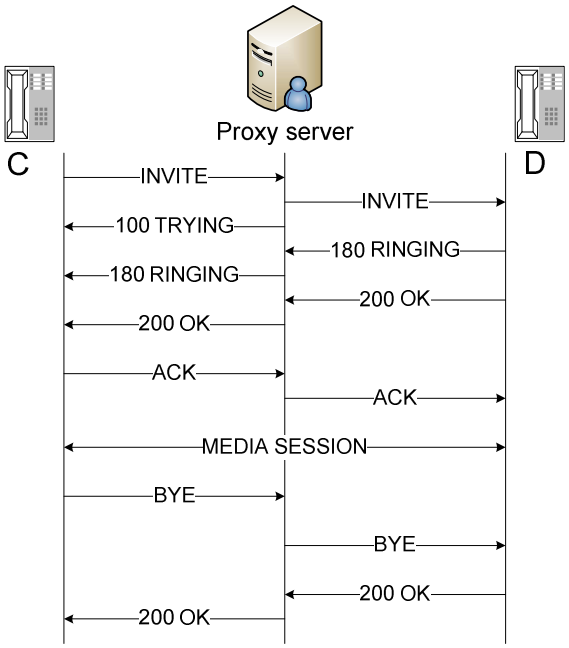
Zprávy jsou buď žádosti nebo odpovědi. Žádostí je nemalý počet např. INVITE (sestavení relace), ACK (potvrzení), BYE (ukončení relace), apod. Odpovědi se značí trojmístným číslem:

- 1xx – Prozatímní nebo informativní charakter
- 2xx – Úspěch
- 3xx – Přesměrování
- 4xx – Značí chybu na straně klienta
- 5xx – Značí chybu na straně serveru
- 6xx – Značí globální chybu



Obr. 12 Sestavení relace bez serveru

Sestavení relace bez serveru je zachyceno na obr. 12. Nejdříve účastník, který spojení inicializuje (v našem případě účastník C), posílá žádost INVITE, tímto žádá účastníka D o spojení. Odpovědi od účastníka D jsou následující: 100 (TRYING – tím účastníkovi C oznamuje, že se zkouší spojit s D), 180 (RINGING – účastníkovi C se ozve ve sluchátku vyzváněcí tón) a 200 (OK – účastník D zvednul sluchátko, neboli D přijal spojení). Dále účastník C vysílá potvrzení (ACK). A nyní může začít rozhovor.



Obr. 13 Komunikace s využitím proxy server

Průběh komunikace s využitím proxy serveru je zachycena na obr. 13. Sestavení relace má stejný průběh jako v předchozím případě s tou výjimkou, že proxy server funguje jako prostředník a přeposílá žádosti a odpovědi. Po ukončení relace jeden z účastníků zavěsí (v našem případě C) a komunikaci ukončí zprávou BYE a následně na ní dostane odpověď 200 (OK – D potvrzuje ukončení – D také zavěsil).

5.3 Softwarové ústředny

Hlasová komunikace po IP sítích je dnes již běžnou záležitostí. Ústředny bývají často řešeny hardwarově. Ovšem efektivnější a leckdy jednodušší je použít IP PBX (Internet Protocol Private

Branch eXchange, česky softwarové ústředny pro IP sítě). Ty dnes do jisté míry nahrazují stávající ústředny.

Ústředna Asterisk používá vlastní protokol IAX2 (již bylo zmíněno na začátku kapitoly 5). I přesto tato ústředna dokáže plně uspokojit uživatele. Asterisk podporuje jak audio tak video rozhovory, nastavení vyzváněcích skupin, podpora faxu, hlasovou schránku aj.

Příkladem freewarových ústředen může být sipXecs (volně ke stažení na sipx-wiki.calivia.com), tato ústředna pracuje na OS Linux. Jak už také název napovídá, tato ústředna používá protokol SIP.

5.3.1 3CX

Dále je známá ústředna od firmy 3CX. Tato ústředna je speciálně navržena specifiky pro Microsoft Windows a využívá protokol SIP. Také nemá velké hardwarové požadavky. 3CX není freewarový program, ale lze volně stáhnout (www.3cx.cz) a má jediné omezení – počet souběžně uskutečňovaných hovorů je maximálně osm [13].

Při instalaci si program umí vytvořit vlastní webový server, ale při použití IIS (Internet Information Services, česky internetová informační služba) výrobce zaručuje bezchybný provoz [14].

3CX nepotřebuje speciální datové rozvody (využívá stávající IP sítě). Konfigurace se provádí pomocí webového konfiguračního rozhraní. Ústředna podporuje video i audio komunikaci. Umí propojit stávající telefonní linky PSTN (Public Switched Telephone Network) se SIP klienty. Také podporuje vyzváněcí skupiny, fax, zasílání upomínek na e-mail, recepčního (spojovatel) a mnoho dalších věcí.

3CX má funkci „firewall checker," která pomáhá nastavit správně firewall.

Ústředna 3CX je výborná volba pro spravování hovorů SIP na OS Microsoft Windows. Tato ústředna v sobě skloubila stabilitu a možnost snadného spravování. A proto byla 3CX vybrána pro tuto práci jako softwarová ústředna.

5.4 Klienti

Klienti mohou být buď softwaroví (pouze se nainstalují do počítače) nebo hardwaroví (zařízení s příslušnými funkcemi).

Softwarový klient je příslušný program, ve kterém se po instalaci nastaví příslušné parametry, a ten pak funguje jako obyčejný telefon. Samozřejmě tento druh klienta má jisté výhody: lehce se modernizuje, není potřeba speciální zařízení, podporuje řadu funkcí (např. seznam kontaktů), aj. Zástupcem může být např. X-lite, nejnovější verze tohoto programu lze stáhnout zdarma z webu firmy CounterPath.

Hardwarový klient je příslušné zařízení (např. SIP telefon). Jsou dražší než softwaroví klienti, hůř se modernizují, ale zato lépe fungují (u softwarových klientů často nefungují různé funkce). Příkladem klienta může být model 7971 od firmy Cisco.

6 Databázové systémy

6.1 Jmenné služby

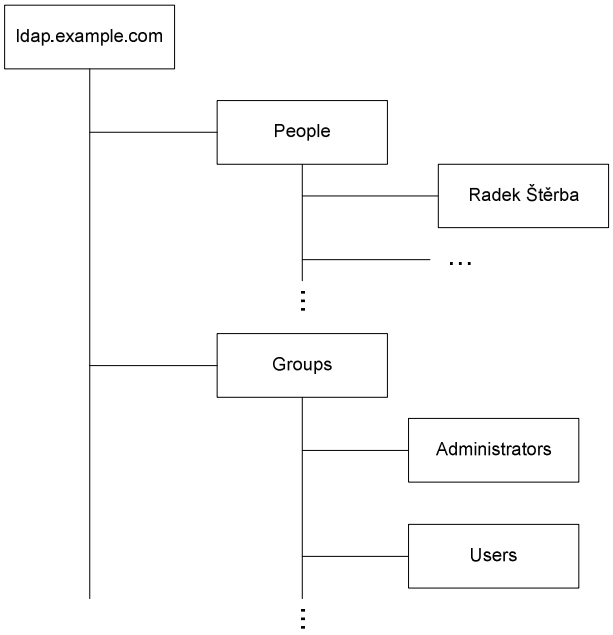
Jmenné služby jsou jedním ze základních pilířů moderních síťových technologií i infrastruktury. Dnes je mnoho entit v komunikačních technologiích, které se identifikují čísly. Snaha je taková, aby se tato čísla dala vyjádřit jmény. Jmenné služby zajišťují vazbu jména na číslo. Důvodem vzniku jmen je zakrytí číselného vzhledu před člověkem (což je pro něj přijatelnější) [15].

Příklad jmenné služby je DNS (má hierarchický charakter a logickou strukturu nezávislou na struktuře IP adres).

6.2 Adresářové služby

Adresářová služba je speciální aplikace pro ukládání strukturovaných dat, organizaci a přístup k nim. Data jsou ukládána ve formě položek a každá položka obsahuje své atributy (specifická data pro danou položku) [15].

Pro návrh struktury adresáře se používá tzv. DIT (Directory Information Tree). Část této struktury je znázorněna na obr. 14. Tato struktura se dá rozdělit na tzv. listy a uzly. Uzly určují větve stromu a listy určují přímo koncové objekty. Pro návrh DIT je celá řada postupů.



Obr. 14 DIT

6.2.1 Jméno položek

DIT a jeho struktura se přímo podepisuje do jména každé položky, toto jméno se označuje DN (Distinguished Name). Toto unikátní jméno je složeno z částí, které přesně popisují jeho polohu v DIT. Příklad takového DN může být následující – `dn: cn=Radek Štěrba, ou=People, dc=lap, dc=example, dc=cz`.

Více o DN lze najít v [15].

6.2.2 Použití

Adresářové služby jsou určeny např. pro návrh komunikačního protokolu. Hlavní myšlenka adresářových služeb je specifikace datového modelu pro ukládání strukturovaných dat a k jejich přístupu [15].

Příkladem adresářové služby je telefonní seznam (seznam lidí, které obsahují různé atributy jako například číslo, adresu a další jiné atributy).

Adresářové služby také dovolují klientům (uživatelům či aplikacím) vyhledávat v databázi podle různých kritérií (např. znám pouze adresu a chtěl bych znát uživatele). Nebo si klient může zažádat o vlastnosti příslušného uživatele [15].

6.3 LDAP

LDAP (Lightweight Directory Access Protocol) je, jak už je vidět z názvu, lehký protokol pro přístup k adresářovým službám (např. ke službám založeným na X.500). Přednosti LDAP jsou hlavně jednoduchost protokolu jako celku a použití rodiny protokolů TCP/IP pro komunikaci.

Postupem času se LDAP osamostatnil a vzniknul i samostatný LDAP server (adresářový server, který používá na komunikaci s klientem protokol LDAP). Dnes se už pod názvem rozumí jak protokol, tak i adresářový server.

Adresář je jako databáze, ale obsahuje více popisných informací. Informace v adresáři se mnohem častěji čtou, než zapisují. Také jsou uspořádány tak, aby hledání ve velkém množství informací bylo rychlé (na to tvůrci LDAP kladou důraz). Je možné, že informace v adresáři jsou uloženy na více místech. Protože vyhledávání v adresáři je rychlé, tak tato segmentace nevadí, pokud je provedena synchronizace těchto míst [16].

6.3.1 Jak funguje LDAP

LDAP je založen na modelu klient – server. Data obsahuje jeden server nebo jsou rozprostřeny mezi více serverů. Když se klient dotáže na jistá data, tak mu je server poskytne nebo ho odkáže na jiný server, který je už má. Pohled na datový adresář se nemění, pokud jsou data umístěna na jednom serveru nebo na více serverech [16].

6.3.2 Vnitřní objekty

Serverový démon se jmenuje `Slapd`. LDAP podporuje různá úložiště dat (databáze). Především to je BDB (Berkeley Data-Base, databáze s podporou transakcí) a LDBM (databáze lehčího typu), SHELL (rozhraní k shellovým skriptům) a PASSWD (tvoří rozhraní pro informace v souboru passwd). BDB je speciálně vytvořeno pro multi-uživatelský přístup.

6.3.3 Konfigurace serveru

Konfigurace probíhá editací konfiguračního souboru `slapd.conf`. Konfigurační soubor obsahuje tři typy informací (globální, strukturové a databázové) [16].

Příklad globální informace je `access to <what> [by <who> <accesslevel> <control>]+` (parametry v závorkách <> mají být nahrazeny textem). Tento příkaz povoluje přístup (určený v `<accesslevel>`) k položkám (`<what>`) uživatelům (`<who>`). Dalším příkazem je `loglevel <integer>` – udává úroveň zaznamenávání ladících příkazů a operačních statistik do systémového logu. Další příkazy jsou `include <filename>`, `idletimeout <integer>` aj.

Další příkazy jsou obecné databázové, ty jsou podporovány všemi databázemi bez výjimky. Nejznámější příkaz je `database <type>` – značí začátek databáze a určuje typ (např. BDB). Další příkaz je `readonly {on | off}`, přepne databázi do režimu „readonly.“ Další příkazy jsou `rootdn <dn>`, `rootpw <password>` aj.

Dále existují databázové příkazy, které jsou specifické pro každý typ databáze (BDB, LDBM, …).

V neposlední řadě jsou také příkazy na řízení přístupu. Ty jsou velmi důležité, neboť povolují nebo naopak odpírají přístup k databázi (čtení i zápis). Nejpoužívanější je příkaz `access to * by * read` (udává, že všichni mohou číst z dané databáze), tento příkaz funguje stejně jako `by anonymous auth`.

Příkazy se mohou lišit (v různé verzi LDAP mohou být jiné). Další popis příkazů lze najít v [16].

6.3.4 Provoz serveru

Server se spouští příkazem `/etc/init.d/slapd start`. A vypíná se analogicky `/etc/init.d/slapd stop`.

Argumenty démona slapd:

- `f <filename>` – Udává alternativní konfigurační soubor pro démona slapd.
- `h <URL>` – Umožňuje zadávat alternativní konfiguraci naslouchání (defaultně je nastavena na `ldap:///`, z toho plyne, že LDAP využívá TCP na každém rozhraní a naslouchá na portě 389). Může se zadat např. dvojice počítač – port (`ldaps://` nebo `ldapi://`).
- `n <service_name>` – Lze zadat jméno služby pro logování (defaultně slapd).
- `l <syslog_local_user>` – Takto se zadává lokální uživatel pro syslog (hodnoty jsou LOCAL0 až LOCAL7).
- `u <user>; -g <group>` – Takto se zadávají uživatelé resp. skupiny, kteří mohou provozovat slapd (zadává se jméno nebo uid resp. gid).
- `r <directory>` – Umožňuje zadat adresář za běhu.
- `d <level>` – Nastavuje se úroveň logování (0 - 2048).

6.3.5 Tvorba databáze

Databázi lze vytvořit dvěma způsoby. Zaprvé je to tzv. způsob on-line. Pomocí LDAP klienta se do databáze přidávají položky (vhodný způsob pro menší databáze). Zadruhé je tvorba databáze off-line. U této metody se využívá speciálních nástrojů, které jsou obsaženy v slapd. Tato metoda je vhodná, pokud je potřeba vložit velké množství položek (řádově tisíce). Nástroje pro tvorbu off-line nepodporují všechny databáze [16].

On-line

Pro vytváření online se obvykle používá nástroj `ldapadd`, který je součástí LDAP (balíčku). Mohou se také použít i speciální klienti (Ldap Browser, Klapd, Directory Administrator, …).

Aby mohl démon slapd zapisovat, musí adresář mít dobře nastavená přístupová práva. Obvykle se tyto práva nastavují pro super-uživatele nebo pro uživatele root. Při konfiguraci databáze se zadají příkazy např. `rootdn <DN>; rootpw <passwd>`. Při tomto zápisu je heslo ve tvaru SHA (Secure Hash Algorithm), ovšem při zápisu dalších hesel (položek v adresáři) lze heslo zapsat jako prostý text (řetězec string). Tímto zápisem se definuje super-uživatel (nemusí být nutně uložen v databázi), který následně bude spravovat databázi.

Příklad zápisu jedné položky by mohl vypadat následovně: `ldapadd -x -W -D "cn=admin, dc=ldap, dc=example, dc=cz" –f ~/new`. Příkaz obsahuje parametry `x` (specifikuje, že se nepoužívá SASL – Simple Authentication and Security Layer), `D` (je použita identita super-uživatele), `W` (program si vyžádá heslo) a `f` (ukazuje, kde je uložen soubor s daty pro import). V souboru „new" jsou uložena data (soupis atributů), která se uloží do adresáře.

Atributy jsou `cn` (Common Name), `uid` (User IDentification), `o` (Organization), `c` (Country), aj. Více atributů lze najít v [16].

Off-line

Tato metoda probíhá off-line (když server není aktivní). Nástroj `slapadd` čte slapd a soubor ve formátu LDIF (LDAP Data Interchange Format), který obsahuje položky, jenž mají být zapsány [16].

Zápis příkazu má následující tvar: `slapadd –l <input_file> -f <slapd_config_file> [-d <debug_level>] [-n <integer>| -b <suffix>]`.

Použité parametry:

- `l <input_file>` – Určení vstupního souboru ve formátu LDIF.
- `f <slapd_config_file>` – Určení konfiguračního souboru slapd.
- `d <debug_level>` – Nastavuje se úroveň logování (0 - 2048).
- `n <integer>` – Udává, která databáze bude modifikována (databáze se rozlišují čísly).
- `b <suffix>` – Udává, která databáze bude modifikována. Číslo databáze se určí porovnáním přípony s příkazem pro příponu databáze (nesmí se použít s parametrem `n` zároveň).

Také existuje příkaz pro vypsání databáze do souboru formátu LDIF. Tento příkaz je `slapcat`. Parametry i jejich význam je stejný jako u `slapadd`.

LDIF

Data ve formátu LDIF mají textový tvar. Různé hodnoty atributů se zapisují na samostatný řádek.

Zápis může vypadat následovně:

```
# uzivatel 12
dn: uid=radek, ou=People, dc=ldap, dc=example, dc=cz
uid: radek
cn: radek sterba
objectClass: person
objectClass: organizationalPerson
objectClass: top
objectClass: shadowAccount
telephoneNumber: 1005
sn: admin ejabberd win32
userPassword: 123456
```

První řádek začíná znakem #, tudíž celý řádek je poznámka. Dále jsou uvedeny atributy DN, UID, CN. Pak také zařazení položky do tříd, dále je taky zadáno telefonní číslo, popis položky (předposlední řádek) a na posledním řádku je uvedeno heslo uživatele (nezašifrované).

6.3.6 Nástroje

Kromě již zmíněných nástrojů LDAP disponuje také nástroji `ldapdelete`, `ldapmodify` a `ldapsearch`.

`Ldapsearch` je nástroj pro vyhledávání v databázi. Příklad příkazu může vypadat následovně: `ldapsearch –b dc=lap, dc=example, dc=cz`. Tento příkaz vypíše obsah celé databáze (parametr `b` určuje počáteční bod hledání). Pokud se zadá klíčové slovo `filter`, tak lze filtrovat hledání v databázi.

`Ldapdelete` slouží k vymazání prvků z databáze. Příkaz může mít tvar: `ldapdelete –W –D "cn=admin, dc=ldap, dc=example, dc=cz" "uid=radek, ou=People, dc=ldap, dc=example, dc=cz"`. Nejdříve se identifikuje super-uživatel a následně se zvolí položka, která má být smazána.

`Ldapmodify` umožňuje,jak již název napovídá, modifikovat položky v adresáři. Příkaz může vypadat: `ldapmodify –f ~/uprava`. V souboru úprava je uloženo, jak a co má být uloženo. Příkaz `ldapadd` je pouze odkaz na příkaz `ldapmodify` s parametrem `a` (přidat novou položku).

Příklad souboru „uprava“:

```
dn: uid=radek, dc=lap, dc=example, dc=cz
changetype: modify
replace: telephoneNumber
telephoneNumber: 1006
```

Takto se v položce „radek“ změní telefonní číslo na hodnotu 1006.

Bližší informace k nástrojům LDAP lze najít v [16] nebo v příslušných manuálových stránkách.

Existují i další nástroje např. migrační nástroje. Tento nástroj je od firmy PADL Software. Používá se ke konverzi konfiguračních souborů do formátu LDIF (kompatibilní s LDAP serverem). Také je tento nástroj vhodný pro migraci uživatelů, skupin uživatelů, počítačů atd. do formátu LDIF.

6.3.7 Autentizace

LDAP server využívá řízení přístupu. Server určuje, co který klient může vidět a dělat (až po autentizaci). Bez autentizace nemá klient přístup ke službám LDAP (kromě anonymní autentizace).

Navázání (bind) je operace pro autentizaci v LDAP. Existují tři typy autentizace: anonymní, jednoduchá a SASL. Pokud server přijme požadavek bez navázání, tak příchozí spojení je od anonymního klienta. Jednoduchá autentizace znamená, že klient pošle nezašifrované DN a heslo klienta. Zprávy SASL jsou žádosti a odpovědi, které přenáší data, jejichž účel je autentizace uživatele a ustanovení bezpečné vrstvy, v níž probíhá pozdější komunikace.

6.3.8 Grafické nástroje

Nástroje pro příkazovou řádku nejsou jedinou cestou jak spravovat LDAP. Existují také grafické nástroje na správu LDAP.

Kldap je grafický klient na správu LDAP, určený pro prostředí KDE (K Desktop Environment). Pomocí tohoto programu lze prohlížet celý adresář. Program KdirAdm je také napsán pro KDE a je určen pro správu adresářů LDAP.

Directory Administrator je hojně požívaný program pro správu adresářů (pro prostředí GNOME). Dále se používá klient GQ, který je napsán pro GNOME, ale funguje i pod KDE. Také lze použít LDAP Browser/Editor.

Nejjistější je ovšem použít nástroje pro příkazovou řádku, ty totiž fungují vždy. Grafické nástroje nejsou většinou od tvůrců LDAP a tudíž nemusí být kompatibilní na 100%.

6.3.9 Závěr

LDAP byl vybrán z důvodů relativní jednoduchosti jak správy tak i nastavení. Pod příslušným OS je rychlý a spolehlivý. Navíc tento program je freeware. Podporuje zabezpečení hesel. Dále podporuje zabezpečený přenos dat (SASL). Z těchto důvodů byl vybrán LDAP pro tuto práci jako adresářový server.

7 Praxe

7.1 LDAP

Ještě před vlastní instalací LDAP je potřeba si nainstaloval OS Linux např. Ubuntu a je doporučeno si stáhnout aktualizace.

Následně se stáhne a nainstaluje LDAP server příkazem `sudo apt-get install slapd ldap-utils` (příkaz platí pro distribuce Debian a z něho odvozené). K tomuto balíčku se přidá migrační nástroj (zmíněn v kap. 6.3.6). Migrační nástroj se nainstaluje příkazem `sudo apt-get install slapd ldap-utils migrationtools` (používá se `sudo`, protože je potřeba mít práva uživatele root).

7.1.1 Základní nastavení

V následném kroku se server LDAP nastaví analogicky podle [17].

Rekonfigurace se spustí příkazem `dpkg-reconfigure slapd`. Následně se nastaví příslušné parametry: doménu serveru např. na ASTERISK3.utko.feec.vutbr.cz, jméno organizace na VUTBR, administrátorské heslo a nakonec typ databáze na BDB.

Dále se upraví konfigurační soubor migračního nástroje `migrate_common.ph` – přepíší se v něm implicitní parametry na:

```
$DEFAULT_MAIL_DOMAIN = “ASTERISK3.utko.feec.vutbr.cz”;
$DEFAULT_BASE = “ASTERISK3.utko.feec.vutbr.cz”;
```

Nyní se zmigrují uživatele v počítači a jejich hesla do formátu LDIF:

```
.migrate_group.pl /etc/group ~/group.ldif
.migrate_passwd.pl /etc/passwd ~/passwd.ldif
```

Ovšem před vložením položek do databáze se musí nejdříve vytvořit skupiny. Ty se vytvoří tak, že se vytvoří soubor `people_group.ldif` s následujícím textem:

```
dn: ou=People, dc=ASTERISK3, dc=utko, dc=feec, dc=vutbr, dc=cz
ou: People
objectclass: organizationalUnit

dn: ou=Group, dc=ASTERISK3, dc=utko, dc=feec, dc=vutbr, dc=cz
ou: Group
objectclass: organizationalUnit
```

Nyní je vše připraveno pro import uživatelů do databáze. Import se provede sekvencí příkazů:

```
ldapadd –x –W –D “cn=admin,dc=ASTERISK3,dc=utko,dc=feec,dc=vutbr,dc=cz” –f
~/peole_group.ldif
ldapadd –x –W –D “cn=admin,dc=ASTERISK3,dc=utko,dc=feec,dc=vutbr,dc=cz” –f
~/group.ldif
ldapadd –x –W –D “cn=admin,dc=ASTERISK3,dc=utko,dc=feec,dc=vutbr,dc=cz” –f
~/people_group.ldif
```

Následně bude založena další skupina pro uživatele, které se budou později vkládat. Tato skupina se pojmenuje USERS a její vytvoření je stejné jako vytvoření skupiny People nebo Group (vytvoří se soubor LDIF s příslušným obsahem a následně se vloží příkazem ldapadd).

Teď lze vytvořit uživatele. Soubor LDIF pro vytvoření uživatele bude mít následující obsah:
dn: uid=tomas, ou=USERS, dc=ASTERISK3,dc=utko,dc=feec,dc=vutbr,dc=cz

uid: tomas
cn: tomas
objectClass: Person
objectClass: organizationalPerson
objectClass: top
objectClass: shadowAccount
telephoneNumber: 1003
sn: user 1003
userPassword: 1003

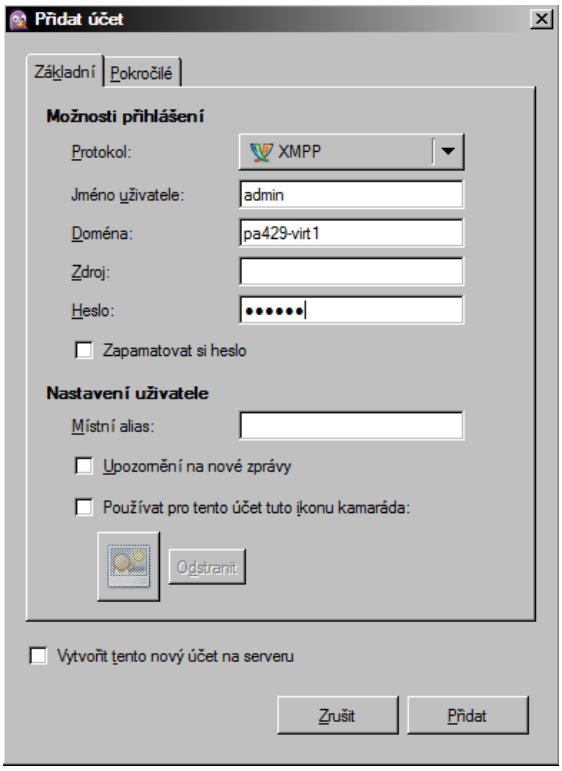
Byl vytvořen další uživatel. Také se musí povolit komunikace na firewallu. K tomu se může použít např. grafická utilita firestarter (`sudo apt-get install firestarter`).

7.2 Jabber server

Do počítače s OS Microsoft Windows XP SP3 (Service Pack 3) se nainstaluje server pro jabber Ejabberd 2.0.3.

7.2.1 Instalace a konfigurace

Server se spustí resp. zastaví souborem `Start ejabberd` resp. `Stop ejabberd` (C:\Program Files\ejabberd-2.0.3\bin). Dále bude vyzkoušena funkčnost serveru a bude připojen k serveru klient (např. Pidgin 2.5.5). V liště programu se zvolí Účty -> Spravovat dále přidat účet. Nastavení je na obr. 15.



Obr. 15 Pidgin, nastavení

Pokud se klient připojí, tak je server nainstalován správně.

Dále je třeba upravit konfigurační soubor `ejabberd.cfg` (C:\Program Files\ejabberd-2.0.3\conf). Do tohoto souboru se vepíše následující text:

```
{auth_method, [internal, ldap]}.
{ldap_servers, ["ASTERISK3.utko.feec.vutbr.cz"]}.
```

```
{ldap_uidattr, "uid"}.
{ldap_base, "dc=ASTERISK3,dc=utko,dc=feec,dc=vutbr,dc=cz"}.
```

První řádek znamená, že je povolena interní autentizační metoda i metoda LDAP. V interní databázi bude uložen pouze administrátor serveru (admin). A v databázi LDAP budou uloženi ostatní uživatelé. Druhý řádek udává adresu serveru. Třetí řádek udává, který atribut v databázi LDAP nese část JID. Protože je automaticky nastaven anonymní přístup do databáze, tak nemusí být zadáno jméno super-uživatele ani jeho heslo [18].

7.2.2 Změna zápisu uživatelů v LDAP

Po procházení souboru syslog (tam si v Ubuntu ukládá LDAP svoje hlášení) a odposlechnutí komunikace programem Wireshark bude zjištěno, že ejabberd vyhledává v databázi celý JID (tomas@pa429-virt1), takže příslušný uživatel se smaže, protože není k ničemu, a nahradí se jiným.

Pozn.: Aby nebyl nutný přechod mezi počítači, tak se použije program putty pro připojení na počítač s Ubuntu přes příkazovou řádku.

Záznam v databázi se odstraní příkazem:

```
ldapdelete –x –W –D “cn=admin,dc=ATERISK3,dc=utko,dc=feec,dc=vutbr,dc=cz” “uid=tomas, ou=USERS, dc=ASTERISK3,dc=utko,dc=feec,dc=vutbr,dc=cz”
```

Obsah souboru LDIF pro vytvoření položky v databázi bude mít následující tvar:

```
dn: uid=tomas@pa429-virt1, ou=USERS, dc=ASTERISK3, dc=utko, dc=feec, dc=vutbr, dc=cz
uid: tomas@pa429-virt1
cn: tomas@pa429-virt1
objectClass: Person
objectClass: organizationalPerson
objectClass: top
objectClass: shadowAccount
telephoneNumber: 1003
sn: user 1003
userPassword: 1003
```

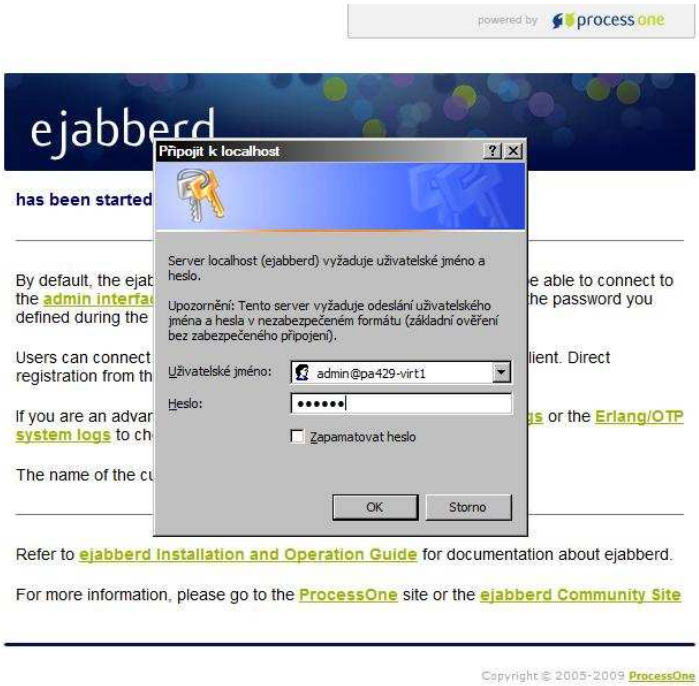
A pak se vloží položka příkazem:

```
ldapadd –x –W –D “cn=admin,dc=ATERISK3,dc=utko,dc=feec,dc=vutbr,dc=cz” –f ~/<název_souboru>.ldif
```

Teď už přihlášení uživatele pomocí klienta pidgin funguje (položka uživatele je pouze v databázi LDAP).

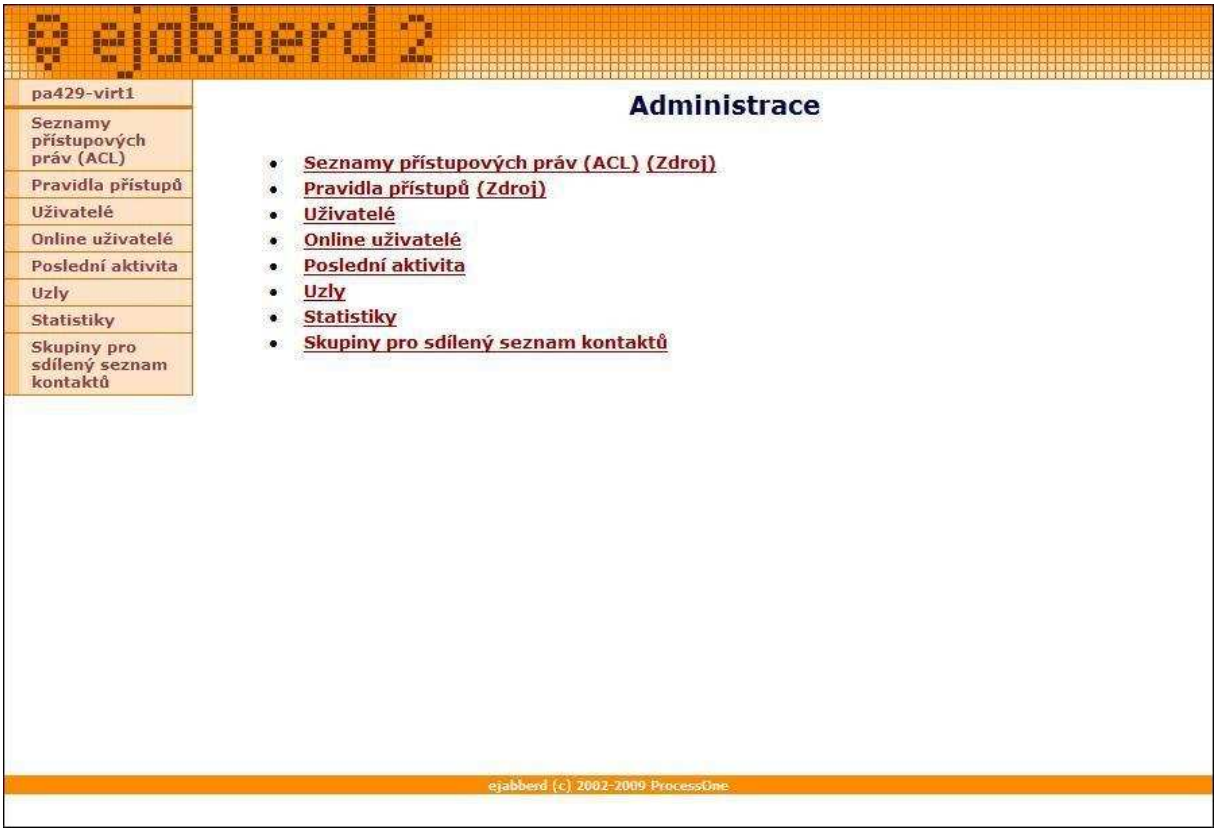
7.2.3 Konfigurační rozhraní

Server ejabberd má příjemné webové konfigurační rozhraní. Při spuštění serveru se automaticky spustí přihlašovací stránka, následně lze kliknout na odkaz `admin interface` a vyplnit příslušné údaje (jméno a doménu oddělené zavináčem a přihlašovací heslo). Příklad přihlášení je na obr. 16.



Obr. 16 Přihlášení na server

Po přihlášení se zobrazí konfigurační rozhraní, kde lze sledovat počet přihlášených uživatelů (kdo konkrétně se přihlásil), lze vidět i zprávy posílané uživatelům, kteří jsou off-line, dále lze tvořit přístupová pravidla v seznamu ACL (Access Contact List) aj.



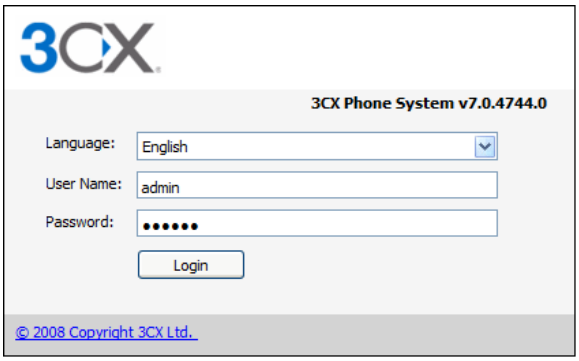
Obr. 17 Webové konfigurační rozhraní

7.3 3CX

Jak již bylo zmíněno v kap. 5.3.1, ústředna 3CX je přizpůsobena pro OS Microsoft Windows. Je doporučeno používat IIS (Internet Information Services) než si nechat vytvořit vlastní webový server. Takže ještě před vlastní instalací je třeba nainstalovat IIS. Instalace IIS se provádí z instalačního CD OS (v OS Microsoft Windows Vista už je IIS součástí systému, není potřeba instalační médium).

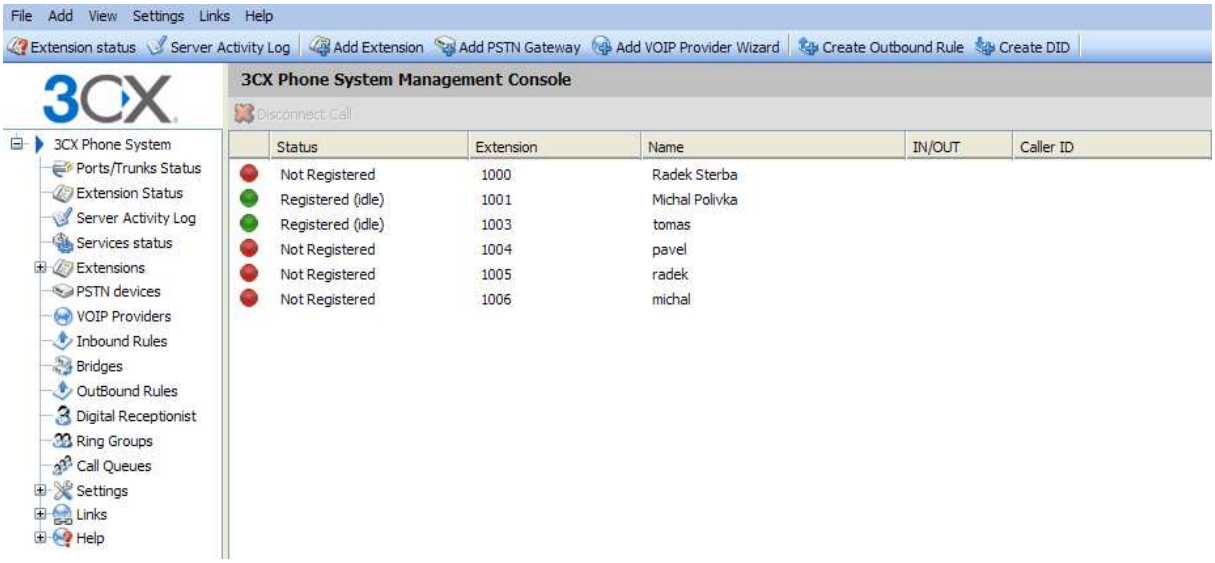
Při instalaci 3CX 7.0 se musí potvrdit licenční smlouva a určí se místo pro instalaci. Poté se potvrdí používání IIS a pak se produkt nainstaluje (pozn.: program si sám nastaví parametry IIS).

Po instalaci se spustí konfigurační wizard. Nejdříve bude zvolen jazyk (English) a pak počet číslic z kterých se bude skládat telefonní číslo (4). Také se může nastavit emailový server (pro zasílání upozornění, …). Dále se zvolí uživatelské jméno a heslo pro přístup do konfiguračního rozhraní. Také je teď vhodné vytvořit alespoň jednoho uživatele, který následně bude zvolen za operátora (k tomuto účelu byli vytvořeni dva uživatelé s čísly 1000 a 1001).



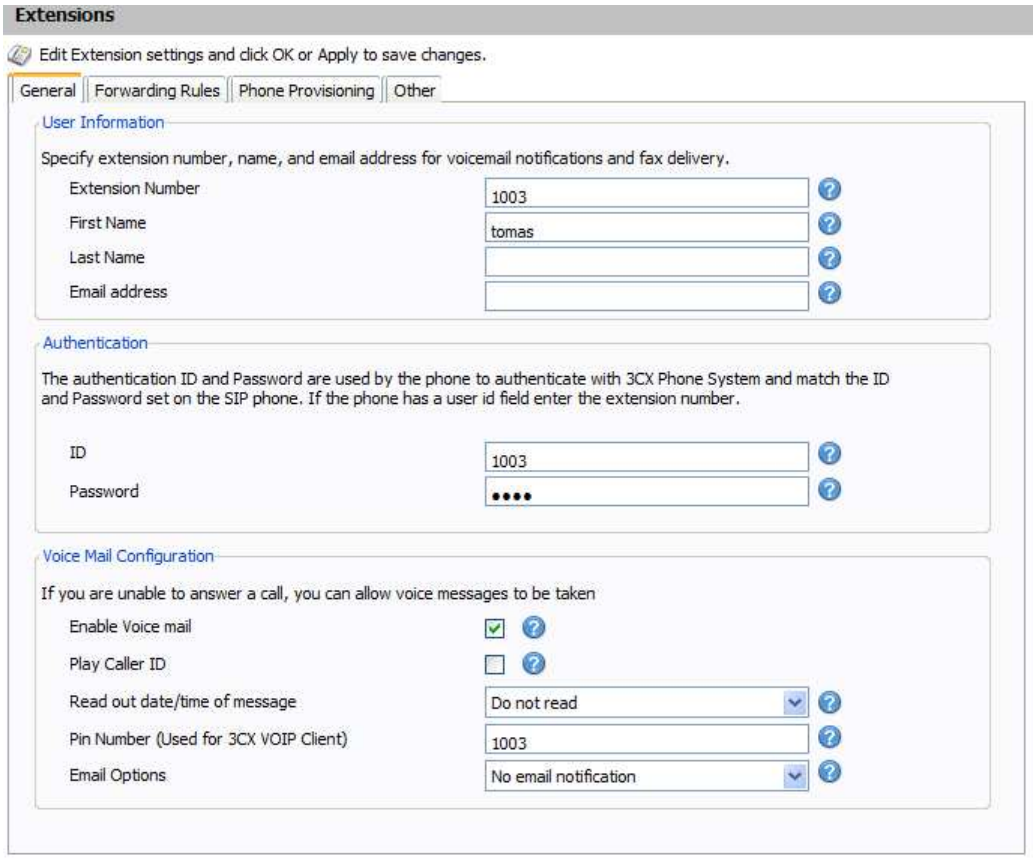
Obr. 18 Přihlášení

K webovému konfiguračnímu rozraní, které je zobrazeno na obr. 19 se lze dostat přes adresu http://localhost/management/MainForm.wgx. Následně se zadá uživatelské jméno a heslo, které bylo nastaveno již dříve.



Obr. 19 3CX webové konfigurační rozhraní

Položka lze přidat přes tlačítko na panelu nástrojů `Add Extension`. Vepíše se pouze číslo položky, jméno a heslo. Následně se tato volba potvrdí a tím se vytvořila položka.

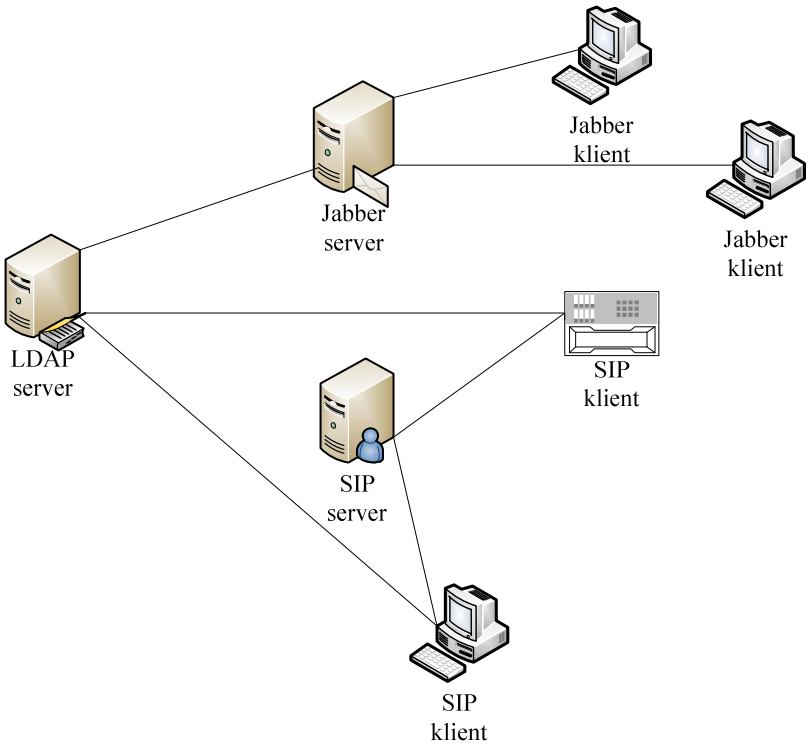


Obr. 20 Vytvoření položky

Nakonec se naváží příslušní klienti na server tak, že se nastaví autorizační jméno (1003), dále heslo a IP adresa serveru (např. X-Lite, klient 3CX se raději nepoužije, protože má problémy s připojováním). Komunikaci s LDAP podporuje např. softphone Ekiga (po zadání adresy LDAP, báze a portu se spojí).

8 Struktura komunikačního systému

Pokud byly nastaveny komponenty tak, jak bylo popsáno v kap. 7, tak funkční komunikační systém bude mít logickou strukturu zobrazenou na obr. 21.



Obr. 21 Komunikační systém

Cílem byla komunikace mezi klienty navzájem a zapojení databáze uživatelů. Komunikaci mezi klienty zabezpečují příslušné servery (SIP server a Jabber server). V serveru LDAP jsou uloženy přihlašovací údaje všech klientů.

Jabber server si čte přímo údaje o klientech ze serveru LDAP, takže si neuchovává vlastní databázi. Výjimkou je administrátor serveru Jabber, který je jediný uložen přímo na serveru Jabber – toto opatření je z bezpečnostních důvodů, např. při pádu serveru LDAP jde spravovat samotný Jabber server.

Požitý SIP server neumí s LDAP serverem komunikovat, takže si musí udržovat vlastní databázi uživatelů. Avšak neteří klienti s LDAP umějí komunikovat (již bylo zmíněno v předcházející kapitole). Takže tito klienti si umí stáhnout informace o ostatních uživatelích a zařadit je do vlastního seznamu kontaktů.

Jediné úskalí této architektury je duplování databáze uživatelů. Avšak tato architektura pracuje efektivněji, než kdyby byl každý server provozován zvlášť. LDAP dovoluje snadnou editaci a správu velkého množství uživatelů. Dále povoluje zapsat četné informace o uživateli. Žádost o poskytnutí informací o uživateli nezatěžuje zbytečně server (jen pro SIP). LDAP také funguje jako centrální správa uživatelských účtů tzn., může být využit pro další aplikace v budoucnu.

V textu byly zmíněny klady a zápory této architektury. Klady jasně převyšují zápory. Tento komunikační systém tedy zefektivňuje komunikaci uvnitř organizace.

9 Závěr

Architektura TPC/IP je hojně rozšířená. V této práci jsou poodhalena úskalí a výhody protokolů TCP (poskytuje spojovanou službu) a UDP (poskytuje nespojovanou službu). Také byla zmíněna adresace, kde a jak se používá. Jedná se především o adresy IPv4.

Tato práce se věnovala studiu IM protokolů (byly vybrány čtyři nejpoužívanější: Jabber, ICQ, Skype a MSN). Byla zde zmíněna různá specifika těchto protokolů (např. otevřenost nebo zabezpečení). Nejlepší volbou je Jabber, protože je otevřený a má volitelné zabezpečení.

Před určením ústředny je seznámení s komunikačními protokoly nezbytnou nutností (SIP, H.323). Jako nejlepší byla zvolena ústředna 3CX, protože je speciálně navržena pro OS Microsoft Windows, je stabilní a nenáročná.

LDAP je pro svoji jednoduchost a nenáročnost hojně využíván. A proto byl popsán on jeho architektura i funkce jako ldapadd – přidávání uživatelů, ldapdelete – mazání uživatelů, formát LDIF a další.

V praktické části je popsáno nastavení všech serverů, klientů a příslušných spojení. A i když při zpracovávání této práce jsem se setkal s OS Linux téměř poprvé a jako administrátor poprvé, tak jsem zvládnul nastavení všech nezbytností (firewall, prohledávání syslogu, instalace balíčků, …).

Práce by mohla pokračovat vytvořením programu nebo vlastního klienta pro VoIP, který by při nastavení statusu „Nepřítomen“ zasílal upozornění na příslušnou IM adresu.

Použitá literatura

[1] **Dostálek, Libor.** *Velký průvodce protokoly TCP/IP: Bezpečnost.* Praha : Vydavatelství a nakladatelství Computer Press, 2003. 80-7220-513-X.

[2] **Macnar, Tomáš.** Síťový protokol TCP/IP. *maturita.cz.* [Online] http://www.maturita.cz/referaty/referat.asp?id=27&pageTitle=S%ED%9Dov%FD%20protokol%20TCP/IP.

[3] **Jalůvka, Dušan.** Volitelné položky IP hlavičky. *www.jaluvka.info.* [Online] 13. 1 2006. http://www.jaluvka.info/projekty/sps/index.html.

[4] **Stuart McClure, Joel Scambray, George Kurtz.** *Hacking bez záhad.* Praha : Grada Publishing, 2007. 5. aktualizované a doplňené vydání. 978-80-247-1502-5.

[5] **Jobra.** Skype. *Připojte se.* [Online] 12. 2 2007. [Citace: 7. 11 2008.] http://www.pripojtese.cz/art_doc-C5CB341B51E075C3C12572740041483B.html.

[6] **Krčmář, Petr.** 10 důvodů proč nepoužívat Skype. *ROOT.CZ.* [Online] 16. 9 2005. [Citace: 7. 11 2008.] http://www.root.cz/clanky/10-duvodu-proc-nepouzivat-skype/.

[7] **Kosek ml., Jiří.** XML. *Vše o WWW.* [Online] 1999. http://www.kosek.cz/clanky/xml/xml-uvod.html.

[8] **Filip, Koval.** Komunikační protokoly. *owebu.cz.* [Online] 6. 9 2007. http://www.owebu.cz/icq/vypis.php?clanek=1127.

[9] **Jelínek, Lukáš.** K čemu je vlastně dobré to SSL? *Aiken.* [Online] 15. 9 2008. http://www.aiken.cz/article/k-cemu-je-vlastne-dobre-to-ssl.

[10] **Číka, Ing. Petr.** *Multimediální služby.* [internet] Brno : VUT v Brně, 2007.

[11] **Bitto, Ondřej.** Jak se volá přes Internet: protokoly H.323 a SIP. *Lupa.* [Online] 24. 1 2007. http://www.lupa.cz/clanky/jak-se-vola-pres-internet-protokoly-h-323-a-sip/.

[12] **Pužmanová, Rita.** Protokol SIP ve zkratce. *Lupa.* [Online] 11. 11 2004. http://www.lupa.cz/clanky/protokol-sip-ve-zkratce/.

[13] **Comservis.** 3CX softwarová telefonní ústředna pro Windows . *Comservis.* [Online] [Citace: 10. 4 2009.] http://www.comservis.cz/index.php?option=com_content&task=view&id=41&Itemid=46.

[14] **3CX.** *Manual 3CX Phone System for Windows Version 7.0.* [internet] Duluth : 3CX, 2008. http://www.3cx.com/manual/3CXPhoneSystemManual7.pdf.

[15] **Sitera, Jiří.** Adresářové služby - úvod do problematiky. *Technická zpráva TEN-155 CZ.* 2000. 4/2000.

[16] **kolektiv.** *Linux Dokumentační projekt, 4. aktualizované vydání.* Praha : Computer Press, a.s., 2008. 978-80-251-1525-1.

[17] How-To set up a LDAP server and its clients. *Debian/Ubuntu Tips & Tricks.* [Online] 22. 2 2007. http://www.debuntu.org/ldap-server-and-linux-ldap-clients.

[18] **Process-One.** Installation and Operation Guide. *Ejabberd open source software.* [Online] http://www.process-one.net/en/ejabberd/guide_en.

[19] **Černohlávek, Ivo.** Protokol TCP/IP a univerzitní síť (1). *Zpravodaj ÚVT MU.* 1992, č.3.

[20] Transportní protokoly TCP/IP - I. *owebu.cz.* [Online] http://pc-site.owebu.cz/?page=PTra.

Seznam použitých zkratek

ACL	(Access Contact List, seznam přístupových práv).
AES	(Advanced Encryption Standard).
AH	(Authentication Header, autentizační záhlaví).
BDB	(Berkeley Data-Base).
C	(Country, země).
CN	(Common Name, společné jméno).
DIT	(Directory Information Tree, informační strom adresáře).
DN	(Distinguished Name, význačné jméno).
DN	(Domain Name, doménové jméno).
DNS	(Domain Name System).
DoS	(Denial of Service, popření služby).
ESP	(Encapsulating Security Payload, stručná bezpečnostní zátěz).
HTML	(HyperText Markup Language).
ICQ	(I seek you, hledám tě).
IETF	(Internet Engineering Task Force, Komise techniky internetu).
IIS	(Internet Information Services, internetová informační služba). ,
IM	(Instant Messaging, rychlé posílání zpráv).
IP	(Internet Protocol).
IP PBX	(Internet Protocol Private Branch eXchange, softwarové ústředny pro IP sítě).
IPv4	(IP version 4, IP verze 4).
IPv6	(IP version 6, IP verze 6).
JID	(Jabber IDentification, identifikace Jabberu).
KDE	(K Desktop Environment).
LAN	(Local Area Network, místní síť).
LDAP	(Lightweight Directory Access Protocol).
LDBM	(databáze lehčího typu).
LDIF	(LDAP Data Interchange Format).
MCU	(Multipoint Control Unit, mutlifunkční jenotka).
MIME	(Multipurpose Internet Mail Extension).
MSN	(The MicroSoft Network, síť Microsoftu).
MTN	(Mini Typing Notification, upozornění na psaní).
O	(Organization, organizace).
OS	(Operation System,operační systém).
PC	(Personal Computer, osobní počítač).
PCM	(Pulse-Code Modulation, pulzně kódová modulace).
PSTN	(Public Switched Telephone Network, stávající telefoní linky).
QIP	(Quiet Instant Pager).
QoS	(Quality of Sevice, kvalita služby).
RSA	(Rivest, Shamir, Adleman).
RTCP	(Real-time Transport Control Protocol).
RTP	(Real-time Transport Protokol).
SA	(Secute Association, bezpečnostní sdružení).
SASL	(Simple Authentication and Security Layer).
SDP	(Session Description Protocol).
SHA	(Secure Hash Algorithm).
SIP	(Session Initiation Protocol).
SP	(Secure Policy, bezpečnostní politika).
SP3	(Service Pack 3, servisní balíček 3).
SPD	(Secure Policy Database, souhrn bezpečnostní politiky).

SPI	(Security Parameters Index, pole bezpečnostních parametrů).
SSL	(Secure Socket Layer).
TCP	(Transmission Control Protocol).
TSL	(Transport Security Layer).
UA	(User Agent, uživatelský agent).
UAC	(User Agent Client, klient uživatelského agenta).
UAS	(User Agent Server, server uživatelského agenta).
UDP	(User Datagram Protocol).
UID	(User IDentification, identifikace uživatele).
UIN	(Universal Internet Number, univerzální internetové číslo).
URI	(Uniform Resource Identifier, jednotný identifikátor zdroje).
USA	(United States of America, Spojené státy americké).
VoIP	(Voice over IP, internetová telefonie).
XML	(eXtensible Markup Language).
XMPP	(Extensible Messaging and Presence Protocol).

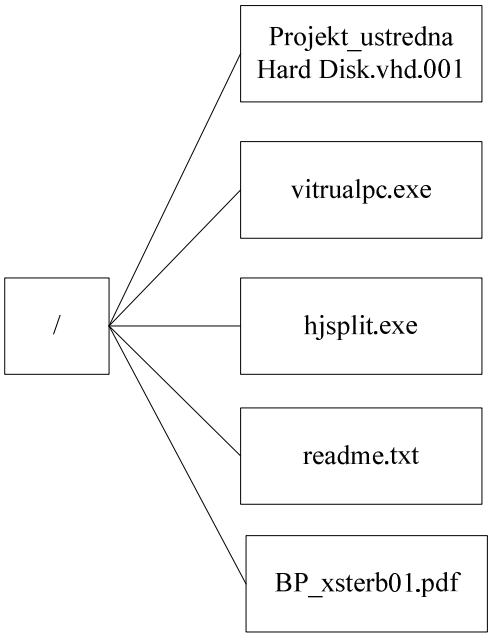
Seznam Příloh

Příloha

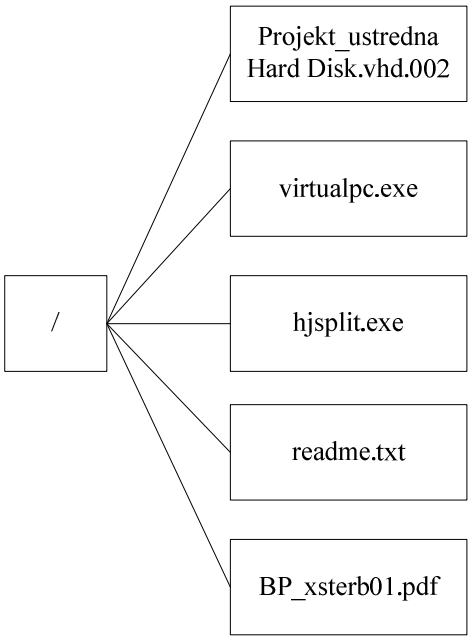
Přiložená média obsahují virtuální počítače (Microsoft Windows XP a Ubuntu), na kterých jsou nainstalovány a nastaveny funkční servery, které byly zmíněny výše, a příslušní klienti. Přístupová jména a hesla jsou uloženy v souboru readme.txt.

Příloha A – obsah přiložených médií

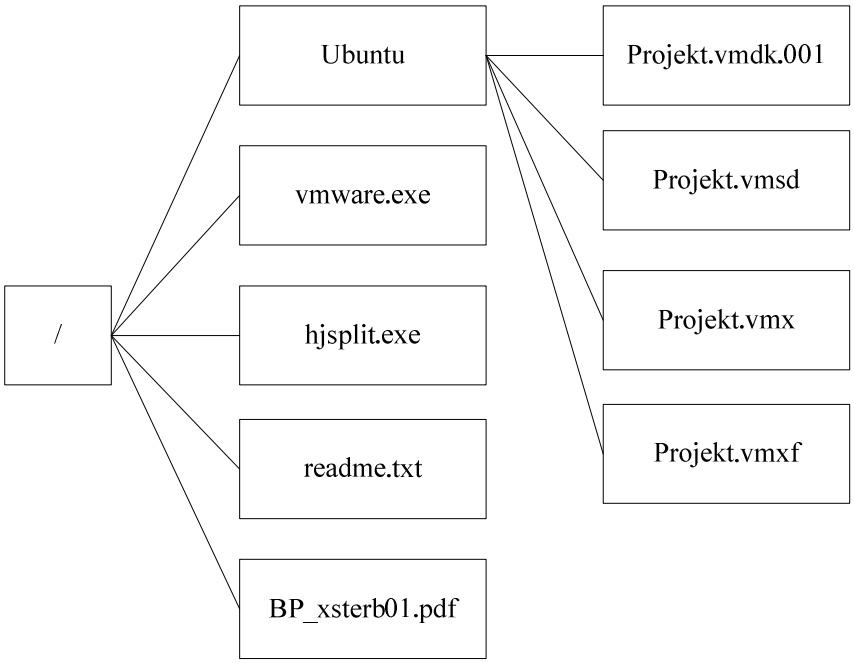
Příloha A.1 Obsah DVD Projekt_XP_01



Příloha A.2 Obsah DVD Projekt_XP_02



Příloha A.3 Obsah DVD Ubuntu_01



Příloha A.4 Obsah DVD Ubuntu_02